# FROG: Fair Removal on Graph

Ziheng Chen
albertchen1993pokemon@gmail.com
Walmart Global Tech, USA

Jiali Cheng
jiali_cheng@uml.edu
University of Massachusetts Lowell, USA

Hadi Amiri
Hadi_Amiri@uml.edu
University of Massachusetts Lowell, USA

Kaushiki Nag
Kaushiki.Nag@walmart.com
Walmart Global Tech, USA

Lu Lin
lulin@psu.edu
Pennsylvania State University, USA

Sijia Liu
liusiji5@msu.edu
Michigan State University, USA

Xiangguo Sun
xiangguosun@cuhk.edu.hk
The Chinese University of Hong Kong, China

Gabriele Tolomei
tolomei@di.uniroma1.it
Sapienza University of Rome, Italy

## Abstract

With growing emphasis on privacy regulations, *machine unlearning* has become increasingly critical in real-world applications such as social networks and recommender systems, many of which are naturally represented as graphs. However, existing graph unlearning methods often modify nodes or edges indiscriminately, overlooking their impact on fairness. For instance, forgetting links between users of different genders may inadvertently exacerbate group disparities. To address this issue, we propose a novel framework that jointly optimizes both the graph structure and the model to achieve *fair* unlearning. Our method rewires the graph by removing redundant edges that hinder forgetting while preserving fairness through targeted edge augmentation. We further introduce a worst-case evaluation mechanism to assess robustness under challenging scenarios. Experiments on real-world datasets show that our approach achieves more effective and fair unlearning than existing baselines.

## CCS Concepts

• **Computing methodologies → Learning paradigms**; **Learning settings**; **Neural networks**; • **Security and privacy → Social network security and privacy**.

## Keywords

Machine Unlearning, Graph Neural Networks, Fairness

**ACM Reference Format:**
Ziheng Chen, Jiali Cheng, Hadi Amiri, Kaushiki Nag, Lu Lin, Sijia Liu, Xiangguo Sun, and Gabriele Tolomei. 2025. FROG: Fair Removal on Graph. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761341

*CIKM '25, Seoul, Republic of Korea.*

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/10.1145/3746252.3761341

## 1 Introduction

Recent breakthroughs in deep learning have significantly advanced artificial intelligence (AI) systems across various domains. In particular, graph neural networks (GNNs) have emerged as a standard approach for addressing graph-related tasks [32, 38], such as node and edge classification – fundamental for applications in social networks (e.g., friend recommendations) and biochemistry (e.g., drug discovery). However, the widespread adoption of GNNs raises concerns about privacy leakage, as training data containing sensitive relationships can be implicitly "memorized" within model parameters. To mitigate the risk of misuse, recent regulatory policies have established the *right to be forgotten* [2], allowing users to remove private data from online platforms. Consequently, a range of graph unlearning methods have been developed to effectively erase specific knowledge from trained GNNs without requiring full retraining.

Although graph unlearning effectively removes edges/nodes, its potential risks – particularly *disparate impact* – are often overlooked. In link prediction, disparate impact refers to disparities in links that stem from sensitive attributes such as gender or race, which are protected under anti-discrimination laws. Recent studies suggest that changes in graph topology, characterized by homophily ratios (see Section 3), can exacerbate bias through feature propagation. For instance, in social networks, removing links to opposite-sex friends may lead to an increased likelihood of users being recommended connections within the same gender group. Thus, long-term accumulation could result in social segregation.

Recently, several algorithms have achieved strong performance in graph unlearning. However, we observed a significant impact on fairness, as edge removal requests alter the graph topology. To examine this effect in social networks, we evaluate two state-of-the-art methods, GNNDelete [9] and GNNCon [36], on Facebook#1684 [20], a social ego network from Facebook app, using gender as the sensitive feature. As shown in Figure 1, both methods fail to maintain dyadic fairness, measured by $\Delta_{DP}$ (see Section 3), when increasing edge removal requests lead to a higher homophily ratio.

The underlying reason is that current algorithms focus solely on designing loss functions to reduce the prediction probability of forgotten edges, without accounting for the bias introduced by

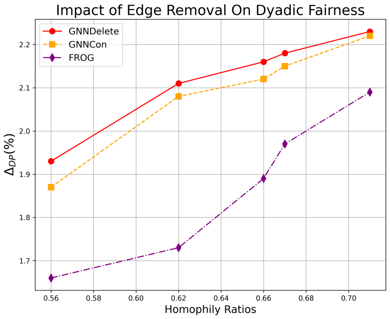

**Figure 1: The impact of removing edges on the fairness of graph unlearning algorithms is shown. The $x$-axis represents the homophily ratio, defined as the proportion of a node's neighbors with identical sensitive features. And the $y$-axis indicates $\Delta_{DP}$, a measure of dyadic fairness.**

edge removal. Moreover, they have also been criticized for *under-forgetting* [9], where an algorithm fails to forget certain edges even after sufficient epochs of unlearning. Consequently, we argue that existing unlearning algorithms do not fully leverage the potential of the graph structure and may not achieve optimal performance.

In this paper, we study a novel and detrimental phenomenon where existing unlearning algorithms can alter the graph structure, inadvertently introducing bias. To address this issue, we propose **FROG** (*Fair Removal on Graph*), a framework designed to effectively forget target knowledge while simultaneously mitigating disparate impact. Our key contributions are as follows:

- **Problem:** We present the first investigation on how graph unlearning impacts graph homophily and disrupts node embeddings through the aggregation mechanism in GNNs, potentially exacerbating discrimination in downstream tasks.
- **Algorithm:** We propose a novel framework, for fair graph unlearning, which integrates graph rewiring and model updating. The graph is rewired by adding edges to mitigate the bias introduced by the deletion request, while removing redundant edges that hinder unlearning. Furthermore, the framework is adaptable to any graph-based unlearning methods for model updates.
- **Evaluation:** In order to truly gauge the authenticity of unlearning performance, we introduce the concept of the "worst-case forget set" in graph unlearning. Experiments on real-world datasets demonstrate that our method improves unlearning effectiveness while mitigating discrimination.

## 2 Preliminaries

***Graph Neural Networks.*** We consider an undirected attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with nodes set $\mathcal{V}$, edge set $\mathcal{E}$ and node features $\mathbf{X}$. Each node is also associated with a categorical sensitive attribute $s_i \in S$ (e.g., political preference, gender), which may or may not be part of its features. The graph topology can be summarized by the adjacency matrix $\mathbf{A}$. Also, we introduce a predictive Graph Neural Network (GNN) model $g_{\boldsymbol{\omega}} : \mathcal{V} \mapsto \mathcal{Y}$, with parameters $\boldsymbol{\omega}$, to predict the nodes' labels as follows:

$$\hat{Y} = f(\mathbf{Z}), \quad \text{with} \quad \mathbf{Z} = g_{\boldsymbol{\omega}}(\mathbf{X}, \mathbf{A}),$$

where $\mathbf{Z}$ represents the node embedding and $\hat{Y} \in \mathcal{Y}$ is the predicted label. The dot product between node embeddings $\mathbf{z}_i^T \mathbf{z}_j$ is used to predict whether edge $e_{ij}$ exists. Also, we refer to $g_{\boldsymbol{\omega}}$ as the "original model" prior to unlearning.

***Graph Unlearning.*** Graph unlearning involves selectively removing certain instances or knowledge from a trained model without the need for full retraining. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and a subset of its elements $\mathcal{G}_f = (\mathcal{V}_f, \mathcal{E}_f, \mathbf{X}_f)$ to be unlearned, we denote the retained subgraph as $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{E}_r, \mathbf{X}_r)$, where $\mathcal{G}_r = \mathcal{G} \setminus \mathcal{G}_f$, with the conditions $\mathcal{G}_f \cup \mathcal{G}_r = \mathcal{G}$ and $\mathcal{G}_f \cap \mathcal{G}_r = \emptyset$. Graph unlearning aims to obtain an unlearned model, denoted as $g_u$, that behaves as if it were trained solely on $\mathcal{G}_r$. Requests for graph unlearning can be broadly categorized into two types: *edge deletion*, where a subset $\mathcal{E}_f \subset \mathcal{E}$ is removed, and *node deletion*, where a subset $\mathcal{V}_f \subset \mathcal{V}$ is removed.

The goal is to derive a new model $g_u$ from the original model $g_{\boldsymbol{\omega}}$ that no longer contains the information from $\mathcal{G}_f$, while preserving its performance on $\mathcal{G}_r$. Since fully retraining the model on $\mathcal{G}_r$ to obtain an optimal model, denoted $g_{\boldsymbol{\omega}^*}$, is often time-consuming, our objective is to approximate $g_{\boldsymbol{\omega}^*}$ by updating $g_{\boldsymbol{\omega}}$ using the unlearning process based on $\mathcal{G}_f$ as follows:

$$g_{\boldsymbol{\omega}} \xrightarrow{\mathcal{G}_f} g_u \approx g_{\boldsymbol{\omega}^*}.$$

***Fairness for Graph Data.*** In this work, we focus on group fairness (also known as disparate impact), which emphasizes that algorithms should not yield discriminatory outcomes for any specific demographic group [31].

For example, in the node classification task, group fairness aims to mitigate the influence of sensitive attributes on individual predictions. Assuming both the target outcome and $S$ are binary-valued, a widely used criterion is *Demographic Parity* (DP) [27]: a classifier satisfies DP if the likelihood of a positive outcome is the same regardless of the value of the sensitive attribute $S$:

$$P(\hat{Y}|S = 1) = P(\hat{Y}|S = 0).$$

In link prediction, we consider the disparity in link formation between intra- and inter-sensitive groups. Extending from *Demographic Parity*, we adopt *Dyadic Fairness* [20], which requires that the predicted likelihood of a link is independent of whether the connected nodes share the same sensitive attribute. A link prediction algorithm satisfies *Dyadic Fairness* if its predictive scores meet the following condition:

$$P(g(u, v)|S(u) = S(v)) = P(g(u, v)|S(u) \neq S(v)).$$

Here, we assume the link prediction function $g$ is modeled as the inner product of nodes' embeddings.

## 3 Motivation

In this section, we present a series of theoretical analyses to elucidate how graph unlearning can lead to unfairness. During the unlearning process, the removal of edges may exacerbate the network homophily, where nodes with similar sensitive features tend to form closer connections than dissimilar ones, inevitably disrupting information flow of graph neural network between nodes within and across sensitive groups.

Inspired by previous work [10, 31], we reveal how the node homophily ratio $\rho$, which is defined as the proportion of a node's neighbors sharing the same sensitive features as the node, can amplify the bias. For simplicity, we focus on a single-layer graph neural network model. Without loss of generality, we assume that node features from two sensitive groups in the network independently and identically follow two different Gaussian distribution $\mathbf{X}^{S_0} \sim \mathcal{N}(\mu^0, \Sigma^0), \mathbf{X}^{S_1} \sim \mathcal{N}(\mu^1, \Sigma^1)$. We proved this theorem in the Appendix.

Theorem 3.1. *Given a 1-layer $g_{\omega}$ with row-normalized adjacency $\tilde{\mathbf{A}} = \mathbf{D}^{-1}\mathbf{A}$ ($\mathbf{D}$ is the degree matrix) for feature smoothing and weight matrix $\mathbf{W}$. Suppose $\exists K > 0, \forall v \in \mathcal{V}, ||\mathbf{X}_v||_2 \leq K$, then the dyadic fairness follows:*

$$\Delta_{DP} = |E_{\substack{(v,u)\\ S_u=S_v}}[\mathbf{z}_v \cdot \mathbf{z}_u] - E_{\substack{(v,u)\\ S_u\neq S_v}}[\mathbf{z}_v \cdot \mathbf{z}_u]| \leq |K \cdot (2\rho - 1)\mathbf{W}\delta|, \quad (1)$$

where $\delta = \mu^0 - \mu^1$, and $\rho$ denotes the homophily ratio defined as:

$$\rho = E_{v\in\mathcal{V}} \frac{|\sum_{u\in N(v)} \mathbb{1}\{S(v) = S(u)\}|}{|N(v)|}.$$

Theorem 3.1 shows that dyadic fairness is bounded by network homophily $\rho$. As $\rho$ increases, due to edge removal requests between nodes with different sensitive attributes, $\Delta_{DP}$ may get enlarged. Conversely, decreasing $\rho$ by adding edges between such nodes enhances cross-group neighborhood connectivity. This smoothing effect on node representations helps mitigate bias. The theoretical findings motivate our algorithmic design presented in next section.

## 4 Problem Formulation

Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ and a fully trained GNN $g_{\omega}$, our goal is to unlearn each edge $e_{uv} \in \mathcal{E}_f$ from $g_{\omega}$, where $\mathcal{E}_f$ denotes the edges to be removed, while mitigating the bias introduced by this removal. Note that node removal can be interpreted as removing all connected edges. The graph structure $\mathcal{E}$ and the GNN $g_{\omega}$ together determine the node embeddings $\mathbf{Z}$, which in turn affect both prediction accuracy and representation fairness. There is broad evidence in literature that graph topology has fundamental effect on the representation [20, 31]. Therefore, we aim to simultaneously obtain an unlearned model $g_u$ and an optimal graph structure $\mathbf{A}^*$. More formally, the task of fair graph unlearning can be cast as:

$$\begin{aligned} g_u, \mathbf{A}^* = &\arg\min_{g,\hat{\mathbf{A}}} \mathcal{L}_{\text{un}}(g, \hat{\mathbf{A}}, \mathbf{X}) + \alpha\mathcal{L}_{\text{fair}}(g, \hat{\mathbf{A}}, \mathbf{X}) \\ \text{subject to: } &||\mathbf{A} - \mathbf{A}^*||_0 \leq N. \end{aligned} \quad (2)$$

The first component ($\mathcal{L}_{\text{un}}$) is an unlearning loss designed to reduce the memorization of forgetting edges $\mathcal{E}_f$, while preserving performance on $\mathcal{E}_r$. Notably, our approach is built to integrate seamlessly with any graph-based unlearning method, allowing any differentiable unlearning loss $\mathcal{L}_{\text{un}}$ to be incorporated in a plug-and-play manner. The second component ($\mathcal{L}_{\text{fair}}$) penalizes violations of representation fairness. In addition, $\alpha$ serves as a scaling factor to trade off between $\mathcal{L}_{\text{un}}$ and $\mathcal{L}_{\text{fair}}$. The detailed form of these losses will be introduced in the following section.

Besides, to avoid omitting much information from $\mathbf{A}$, we discourages $\mathbf{A}^*$ to be too far away from $\mathbf{A}$ by limiting the number of edges to be modified, i.e., to a maximum of $N$ edges. Here we adopt $L^0$-norm to quantify the distance between $\mathbf{A}^*$ and $\mathbf{A}$.

## 5 FROG: Fair Removal on Graph

Directly solving Equation 2 is a non-trivial task, particularly when $\mathcal{L}_{\text{un}}$ and $\mathcal{L}_{\text{fair}}$ are in conflict with each other, leading to an imbalance of both objectives. To address this, we formulate the process as a Stackelberg game [30], or leader-follower game. First, the leader performs *fair edge augmentation* to recover the fairness degradation caused by unlearning requests. Then, the follower performs *sparse structure unlearning* to achieve the unlearning objective. By alternatively optimizing these two objectives, we aim at finding a balance between preservation of fairness and effective unlearning. In addition to the GNN parameters, the underlying graph topology is also being optimized, since graph topology has a fundamental effect on graph fairness (see Section 3).

In the upper problem of *fair edge augmentation* taken by the leader, an augmenter $f$ takes an input $\mathbf{A}$ and produces an augmented graph $\mathbf{A}^{\text{aug}} = f(\mathbf{A})$. In the lower problem of *sparse structure unlearning* taken by the follower, a pruner $p$ removes redundant edges from $\mathbf{A}^{\text{aug}}$ to get the optimal graph $\mathbf{A}^* = p(\mathbf{A}^{\text{aug}})$. As shown in Figure 2, leveraging edge augmentation to explore fair structures, subsequently refined through pruning to align with unlearning objectives, increases the potential to escape sub-optimal solutions. These iterative steps can be unified by formulating the problem as the following bi-level optimization:

$$\begin{aligned} g_u, p =&\arg\min_{g,p} \mathcal{L}_{\text{un}}(g, p(\mathbf{A}^{\text{aug}}), \mathbf{X}) + \alpha\mathcal{L}_{\text{fair}}(g, p(\mathbf{A}^{\text{aug}}), \mathbf{X}) \\ \text{subject to: } & f = \arg\min_{f} \mathcal{L}_{\text{fair}}(g, f(\mathbf{A}), \mathbf{X}) \quad \mathbf{A}^{\text{aug}} = f(\mathbf{A}) \\ & ||\mathbf{A} - \mathbf{A}^*||_0 \leq N. \end{aligned} \quad (3)$$

### 5.1 Fair Edge Augmentation

As shown in Section 3, edge removal can increase the homophily ratio, thereby affecting representation fairness among local neighbors. To address this, $f$ targets on adding *inter-group links* within local neighborhoods in $\mathcal{G}_f$ to mitigate biases introduced by edge removal. Specifically, We introduce a Boolean perturbation matrix $\mathbf{M}^{\text{aug}} \in \{0, 1\}$ to encode whether or not an edge in $\mathcal{G}$ is modified. That is, the edge connecting nodes $i$ and $j$ is added, if and only if $\mathbf{M}^{\text{aug}}_{ij} = \mathbf{M}^{\text{aug}}_{ji} = 1$. Given the adjacency matrix $\mathbf{A}$, this process can be described as :

$$\mathbf{A}^{\text{aug}} = \mathbf{A} + (\mathbf{J} - \mathbf{I} - \mathbf{A}) \circ \mathbf{M}^{\text{aug}}. \quad (4)$$

Here $\mathbf{J}$ denotes an all-one matrix. We exclude all edges in $\mathcal{E}_f$ from $\mathbf{A}$ and set the corresponding entries in $\mathbf{M}^{\text{aug}}$ to zero to prevent reintroduction. Due to the discrete nature of $\mathbf{M}$, we relax edge weights from binary variables to continuous variables in the range $(0, 1)$ and adopt the reparameterization trick for gradient-based optimization. Specifically, we sample $\mathbf{M}^{\text{aug}} \sim \text{Bernoulli}(\tilde{\mathbf{M}}^{\text{aug}})$, where $\tilde{\mathbf{M}}^{\text{aug}}$ represents the predicted probabilities of edge additions. For each pair of nodes $(i, j)$, the embeddings $z_i$ and $z_j$ of $g_{\omega}$ are used to estimate the probability of the edge $e_{ij}$. After the $\mathbf{M}^{\text{aug}}$ is sampled, only the *inter-group links* are incorporated. To support the end-to-end training, we leverage the Gumbel-Softmax trick[17] to approximate the non-differentiable Bernoulli sampling:

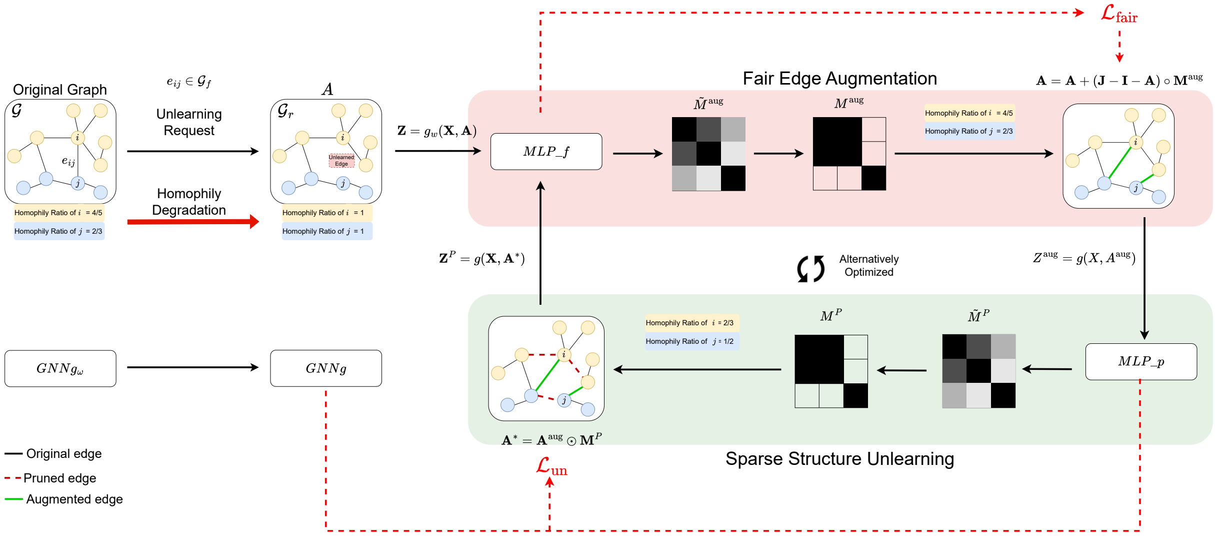
$\mathcal{L}_{\text{fair}}$
Fair Edge Augmentation
$\mathbf{A} = \mathbf{A} + (\mathbf{J} - \mathbf{I} - \mathbf{A}) \circ \mathbf{M}^{\text{aug}}$
Original Graph
$\mathcal{G}$
$e_{ij}$
$i$
$j$
Homophily Ratio of $i$ = 4/5
Homophily Ratio of $j$ = 2/3
$e_{ij} \in \mathcal{G}_f$
Unlearning Request
Homophily Degradation
$A$
$\mathcal{G}_r$
Unlearned Edge
Homophily Ratio of $i$ = 1
Homophily Ratio of $j$ = 1
$\mathbf{Z} = g_w(\mathbf{X}, \mathbf{A})$
$MLP\_f$
$\tilde{M}^{\text{aug}}$
$M^{\text{aug}}$
Homophily Ratio of $i$ = 4/5
Homophily Ratio of $j$ = 2/3
$\mathbf{Z}^P = g(\mathbf{X}, \mathbf{A}^*)$
Alternatively Optimized
$Z^{\text{aug}} = g(X, A^{\text{aug}})$
Homophily Ratio of $i$ = 2/3
Homophily Ratio of $j$ = 1/2
$M^P$
$\tilde{M}^P$
$MLP\_p$
$GNNg_\omega$
$GNNg$
$\mathbf{A}^* = \mathbf{A}^{\text{aug}} \odot \mathbf{M}^P$
Sparse Structure Unlearning
$\mathcal{L}_{\text{un}}$
Original edge
Pruned edge
Augmented edge

**Figure 2: Schematics of FROG: (*i*) Request for edges removal; (*ii*) Adding edges to mitigate bias caused by edge removal; and (*iii*) Removing redundant edges that obstruct the unlearning process.**

$$\widetilde{\mathbf{M}}^{\text{aug}} = \sigma\left(\frac{f([z_i; z_j]) + f([z_j; z_i])}{2}\right)$$
$$\mathbf{M}^{\text{aug}} = \frac{1}{1 + \exp\Big(-(\log(\widetilde{\mathbf{M}}^{\text{aug}}) + \mathbf{G})/\tau\Big)} \tag{5}$$

where $f$ denotes a multi-layer perceptron(MLP), $[;]$ indicates concatenation and $\sigma$ is the sigmoid function. We ensure that $\mathbf{M} = \mathbf{M}^T$ in Equation 5 to maintain the symmetry of the perturbation matrix. Given the predicted probabilities $\widetilde{\mathbf{M}}^{\text{aug}}$, the relaxed Bernoulli sampling yields a continuous approximation, where $\tau$ is a temperature hyperparameter and $\mathbf{G} \sim \text{Gumbel}(0, 1)$ is sampled from the standard Gumbel distribution.

As our objective is to generate fair augmentations by adding edges, the ideal augmenter $f$ targets on finding the optimal structure $\mathbf{A}^{\text{aug}} = f(\mathbf{A})$ that achieves representation fairness. However, we cannot achieve it via supervised training because there is no ground truth indicating which edges lead to fair representation and should be added. To address this issue, we propose to use a contrastive loss to optimize the augmenter $f$.

Inspired by [19, 35, 39], we propose a contrastive loss which explicitly penalizes $\mathbf{A}^{\text{aug}}$ for increasing the edge probability between nodes sharing the same sensitive feature. For clarity, we treat each node $i$ as an anchor and define the following pairs based on its relationship with other samples. Specifically:

- *Intra positive pairs:* refers to pairs of an anchor and its connected nodes that share the same sensitive features. $V^+_{intra}(i) = \{j : \mathbf{A}^{\text{aug}}[i, j] = 1 | S(i) = S(j)\}$.
- *Inter positive pairs:* refers to pairs of anchors and their connected nodes that share different sensitive features. $V^+_{inter}(i) = \{j : \mathbf{A}^{\text{aug}}[i, j] = 1 | S(i) \neq S(j)\}$.
- *Intra negative pairs:* refers to pairs of an anchor and its non-connected nodes that share different sensitive features. $V^-_{intra}(i) = \{j : \mathbf{A}^{\text{aug}}[i, j] = 0 | S(i) = S(j)\}$.

For each anchor, our key idea is to ensure that positive and negative samples share the same sensitive attributes as the anchor, rendering sensitive features uninformative for link probability. We define the $V^-_{intra}(i)$ as negative pairs, while treating $V^+_{intra}(i)$ and $V^+_{inter}(i)$ as positive pairs. Based on this, we design $\mathcal{L}_{\text{fair}}$ to enhance the link probability between the anchor and nodes in positive pairs relative to negative pairs. It is formulated as follows:

$$\mathcal{L}_{\text{fair}} = \sum_{v_i \in \mathcal{V}} \frac{-1}{|V_P(i)|} \sum_{j \in V_P(i)} \log \frac{\exp(z^{\text{aug}}_j \cdot z^{\text{aug}}_i)}{\sum\limits_{k \in V^-_{intra}(i)} \exp(z^{\text{aug}}_i \cdot z^{\text{aug}}_k)} \tag{6}$$

Here $V_P(i) = V^+_{intra}(i) \cup V^+_{inter}(i)$ and $\mathbf{z}^{\text{aug}} = g_\omega(\mathbf{X}, \mathbf{A}^{\text{aug}})$ where $z^{\text{aug}}$ represents the node embedding in $\mathbf{A}^{\text{aug}}$. The $g_\omega$ is fixed during the optimization of $\mathcal{L}_{\text{fair}}$.

## 5.2 Sparse Structure Unlearning

To achieve fair unlearning, we consider finding an optimal structure by eliminating redundant edges from $\mathbf{A}^{\text{aug}}$, while keeping the unbiased and informative ones. Following other approximate-based unlearning methods, we also adopt a learnable mechanism to adjust the original model for the target. Specifically, we learn a pruner over $\mathbf{A}^{\text{aug}}$ to achieve the fair unlearning target. We optimize for the graph adjacency as follows:

$$g_u, p = \arg\min_{g_u, p} \mathcal{L}_{\text{un}}(p(\mathbf{A}^{\text{aug}}), \mathbf{X}) + \alpha \mathcal{L}_{\text{fair}}(p(\mathbf{A}^{\text{aug}}), \mathbf{X}) + \beta \mathcal{L}_{\text{dist}}. \tag{7}$$

Here, we set $\mathcal{L}_{\text{dist}} = ||p(\mathbf{A}^{\text{aug}}) - \mathbf{A}||_1$, where $|| \cdot ||_1$ is the $L^1$-norm, to constrain the number of edge modifications from the original adjacency matrix $\mathbf{A}$. Similar to $f$, the pruner $p$ is an MLP to generate the mask matrix $\mathbf{M}^p$ according to Equation 5.

$$\mathbf{M}^p = \frac{1}{1 + \exp(-(\log \widetilde{\mathbf{M}}^p + \mathbf{G})/\tau)}, \quad \mathbf{A}^* = \mathbf{A}^{\text{aug}} \odot \mathbf{M}^p. \tag{8}$$

Note that $\widetilde{\mathbf{M}}^p$ is constructed using embeddings on $\mathbf{A}^{\text{aug}}$ with $z^{\text{aug}} = g(\mathbf{X}, \mathbf{A}^{\text{aug}})$. Building on this, our method could be seamlessly combined with any graph unlearning loss function as $\mathcal{L}_{\text{un}}$ [9][21][36].

Here we adopt the $\mathcal{L}_{\text{un}}$ from GNNDelete [9], which formulates the unlearning loss into two properties. For each deleted edges $e_{ij}$:

- *Deleted Edge Consistency*, where deleted edges should have a similar predicted probability to randomly sampled unconnected edges.
  $\mathcal{L}_{\text{DEC}} = \mathcal{L}_{\text{MSE}}(\{[z_i^p; z_j^p] | e_{ij} \in \mathcal{E}_f\}, \{[z_i^{\text{aug}}; z_j^{\text{aug}}] | i, j \in \mathcal{V}\})$.
- *Neighborhood Influence*, where node embeddings post-unlearning should be similar to prior-unlearning.
  $\mathcal{L}_{\text{NI}} = \sum \left(\mathcal{L}_{\text{MSE}}(z_u^p, z_u^{\text{aug}}) | u \in \mathcal{S}_{ij}\right)$, where $S_{ij}$ is the local enclosing subgraph of $e_{ij}$ and $\mathbf{Z}^p = g_u(\mathbf{X}, \mathbf{A}^*)$.

Finally, $\mathcal{L}_{\text{un}} = \lambda \mathcal{L}_{\text{DEC}} + (1 - \lambda)\mathcal{L}_{\text{NI}}$; following GNNDelete, we set the optimal $\lambda = 0.5$.

We present a theoretical observation demonstrating how the sparsification operator can facilitate unlearning.

Theorem 5.1. *(Bounding edge prediction of unlearned model $g_u$ by $g_\omega$) Let $e_{ij}$ be an edge to be removed, and $\mathbf{W}$ be the last layer weight matrix in $g_\omega$. Then the norm difference between the dot product of the node representations $z_i, z_j$ from $g_\omega$ and $z_i', z_j'$ from the unlearned model $g_u$ is bounded by:*

$$\left\langle z_i, z_j \right\rangle - \left\langle z_i', z_j' \right\rangle \le (\frac{1}{2}\|\mathbf{W}_D^L\|^2 - 1)\|z_i - z_j\|^2 + \|\mathbf{W}_D^L \mathbf{W}\|^2 \|\Delta\|^2 \tag{9}$$

*where $\Delta = \sum_{k \in C_i} \mathbf{h}_k^{L-1} - \sum_{k \in C_j} \mathbf{h}_k^{L-1}$.*

Here $C_i$ and $C_j$ represent the common neighbors with masked edges connecting to nodes $i$ and $j$, respectively. In GNNDelete, $\mathbf{W}_D^L$ denotes the deletion matrix at layer $L$, and $\mathbf{h}_k^{L-1}$ represents the embedding of node $k$ from the previous layer. Detailed derivations are shown in the Appendix. The first term in the bound ensures the stability of the deletion operator, while the second term suggests that masking edges from common neighbors can enlarge the gap between $g_\omega$ and $g_u$ in predicted probability of $e_{ij}$, thereby enhancing the unlearning capability.

To sum up, the training process can be described as a Bi-level optimization(Equation 3), where $\alpha, \beta$ are hyperparameters. In each iteration, we first update $f$ to minimize $\mathcal{L}_{\text{fair}}$ while keeping $p$ and $g$ fixed, then update both $g$ and $p$. The training continues until convergence. An overview of FROG is illustrated in Figure 2.

### 5.3 FROG-Joint: Baseline

Instead of the Bi-Level optimization, we also introduce a Joint optimization algorithm to solve Equation 2 as a baseline. Following [18], we directly model the $\mathbf{A}^*$ as a function of the embeddings as Equation 5. In this way, the number of parameters for modeling graph structure no longer depends on the number of nodes, hence avoid learning $O(|\mathcal{V}|^2)$ parameters and renders the algorithm applicable to large-scale graphs.

## 6 Worst-Case Evaluation

Inspired by [12], we evaluate unlearning methods with two different challenging settings: ($i$) *worst-case unlearning*, where $\mathcal{G}_f$ consists of edges that are hardest to forget, and ($ii$) *worst-case fairness*, where $\mathcal{G}_f$ consists of edges that negatively impacts fairness on $\mathcal{G}_r$ post-unlearning. Let $w \in \{0, 1\}^{|\mathcal{E}|}$ denote a binary mask over all edges, where $w_{i,j} = 1$ indicates that the edge $e_{ij}$ belongs to the forget set. Our objective is to optimize $w$ such that $\mathcal{G}_f$ contains all hard-to-forget edges or those critical for fairness.

***Worst-case unlearning.*** We select the forget set $\mathcal{G}_f$ to maximize the difficulty of effective unlearning. In other words, after unlearning $\mathcal{G}_f$, the unlearned model will exhibit a low loss on $\mathcal{G}_f$, indicating a failure to fully eliminate the influence of $\mathcal{G}_f$ from the model. Specifically, we solve:

$$\min_{w \in S} \sum_{e_{ij} \in \mathcal{G}} [w_{ij} \mathcal{L}_{\text{LP}}(g_u; z_i, z_j)] + \gamma \|w\|_2^2 \tag{10}$$

$$\text{subject to:} \quad g_u = \arg\min_g \mathcal{L}_{\text{un}}(g; w), \tag{11}$$

where $\mathcal{L}_{\text{LP}}$ is the link prediction loss.

In the upper-level optimization, we search for the edges defined by the binary edge mask $w$ that yield the worst unlearning performance. In other words, the loss on the forget set $\sum_{e_{ij}} \mathcal{G} \in [w_{ij}\mathcal{L}_{\text{LP}}(g_u; z_i, z_j)]$ is minimized (unsuccessful unlearning). We additionally regularize the size of $w$ with the $L_2$ norm, since unlearning requests are much sparser than the original dataset. *In the lower-level* optimization, the unlearned model $g_u$ is obtained based on the forget set selected by $w$.

***Worst-case fairness.*** We choose the forget set $\mathcal{G}_f$ to maximize fairness degradation. That is, after unlearning $\mathcal{G}_f$, the $\mathcal{L}_{\text{fair}}$ on the retained set $\mathcal{G}_r$ is maximized, indicating a failure to preserve the fairness. We solve:

$$\max_{w \in S} \sum_{e_{ij} \in \mathcal{G}} [(1 - w_{ij})\mathcal{L}_{\text{fair}}(g_u; z_i, z_j)] + \gamma \|w\|_2^2 \tag{12}$$

$$\text{subject to} \quad g_u = \arg\min_g \mathcal{L}_{\text{un}}(g; w). \tag{13}$$

## 7 Experiments

To evaluate the effectiveness of our proposed model, we examine the following questions:

- **RQ1:** How is the unlearning efficacy of FROG and its impact on graph fairness under uniform cases?
- **RQ2:** How is the unlearning efficacy of FROG and its impact on graph fairness under worst-case scenarios?

***Datasets.*** We perform experiments over the following real-world datasets: CiteSeer [1], Cora [1], OGB-Collab [9], Facebook#1684. [20] and Pubmed [9]. Facebook#1684 is a social ego network from the Facebook app, and we select gender as the sensitive feature. The rest citation networks, each vertex represents an article with descriptions as features. A link stands for a citation. We set the category of an article as the sensitive attribute. These datasets usually serve as a commonly used benchmark datasets for GNN performance over link prediction and node classification tasks.

***Baselines.*** We compare FROG to the following baselines: ($i$) Retrain, which refers to training from scratch; ($ii$) GA [15], which performs gradient ascent on $\mathcal{G}_f$; ($iii$) GER [4], a re-training-based machine unlearning method for graphs; ($iv$) GNNDelete [9], an approximate graph unlearning method that treats deleted edges as unconnected node pairs; ($v$) GNNCON [36] a contrastive learning

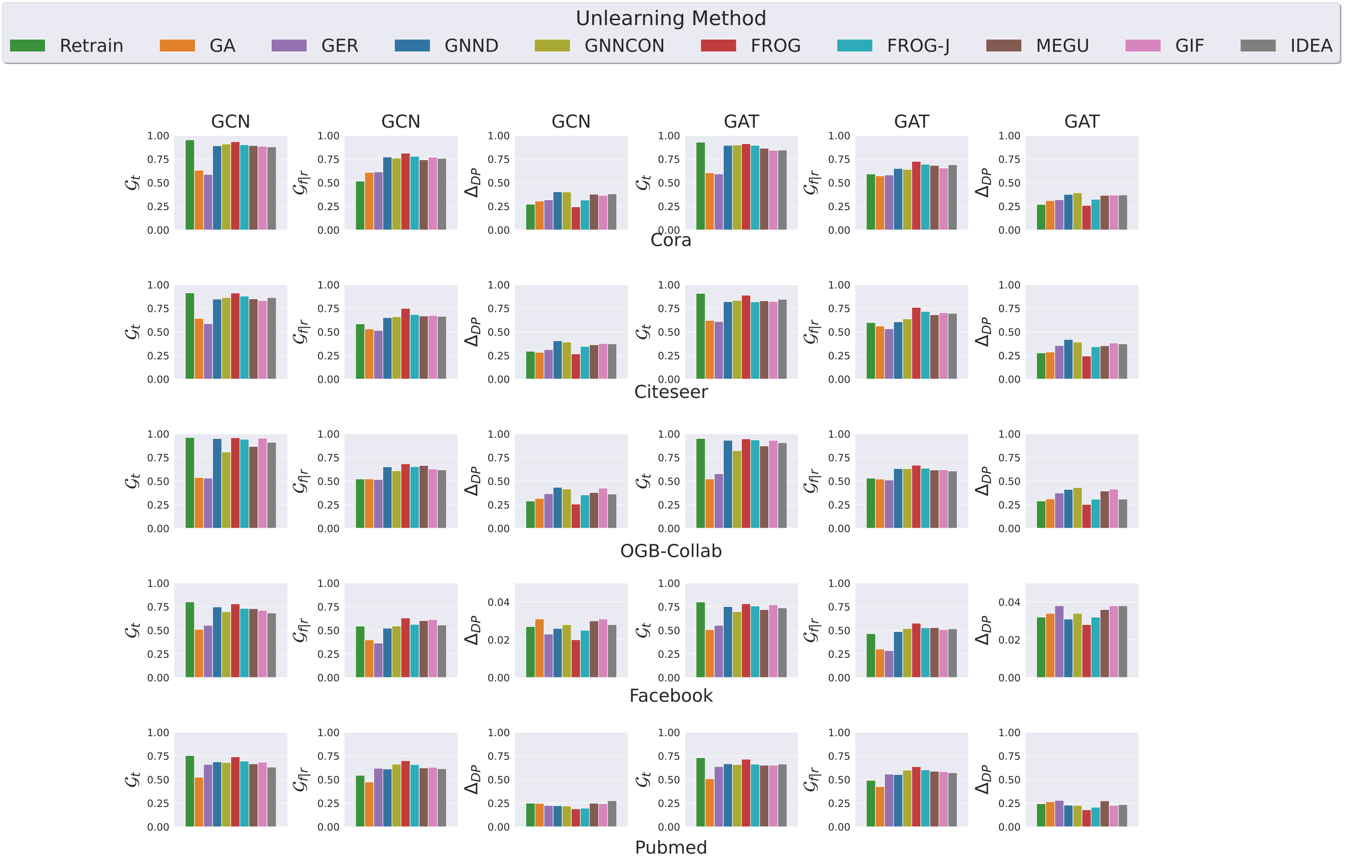

**Figure 3: Effectiveness and fairness performance of edge unlearning on Cora, CiteSeer, OGB-Collab, Facebook, and Pubmed.**

based method. Finally, (*vi*) MEGU [21], (*vii*) GIF [33], and (*viii*) IDEA [11], which represent advanced graph unlearning methods.

***Unlearning Task.*** We evaluate FROG under two unlearning tasks: (*i*) *node unlearning*, where a subset of nodes $\mathcal{N}_f \in \mathcal{N}$ and all their associated edges are unlearned from $g_w$; and (*ii*) *edge unlearning*, where a subset of edges $\mathcal{E}_f \in \mathcal{E}$ are unlearned from $g_w$. In line with prior works, the forget set comprises 5% of the entire dataset.

***Sampling of Forget Set.*** For worst-case unlearning, the forget set $\mathcal{G}_f$ is chosen through optimization according to 10 and 12. For uniform cases, the forget set $\mathcal{G}_f$ is randomly selected within 3-hops of $\mathcal{G}_t$. Due to the limited space, we only conduct edge unlearning in the worst-case evaluation.

***Evaluation Metrics.*** We evaluate the unlearned model's performance from the following two perspectives:

- *Effectiveness-oriented*, which probes if $\mathcal{G}_f$ is unlearned from $\mathcal{G}_w$ while preserving model utility. Specifically, we compute (*i*) the test set AUROC $\mathcal{G}_t(\uparrow)$, (*ii*) the forget-retain knowledge gap $\mathcal{G}_{f|r}(\uparrow)$ [6, 9] which quantifies how well a model distinguishes unlearned and retained data. Specifically, the knowledge gap is computed as the AUROC score with the prediction logits of $\mathcal{G}_r$ and $\mathcal{G}_f$, and their labels as 1 and 0, respectively. We also demonstrate the success rate of membership inference attack (MIA) by MI(↑).

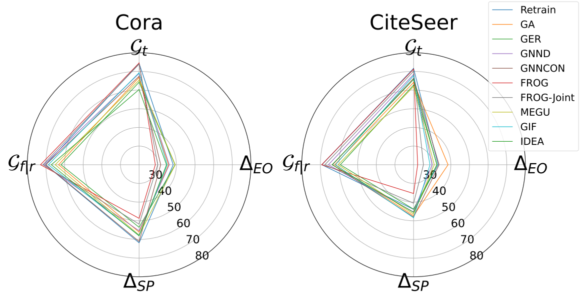

**Figure 4: Effectiveness and fairness performance of node unlearning.**

- *Fairness-oriented*, which evaluates the fairness of the unlearned model. In node classification, we focus on Equalized Odds (EO) $\Delta_{EO} = |p(\hat{y} = 1|y = 1, s = 1) - p(\hat{y} = 1|y = 1, s = 0)|$ and Statistical Parity (SP) $\Delta_{SP} = |p(\hat{y} = 1|s = 1) - p(\hat{y} = 1|s = 0)|$ [27] to evaluate the group disparity. In the link prediction scenario, we directly use Demographic Parity $\Delta_{DP}$ to measure the dyadic fairness.

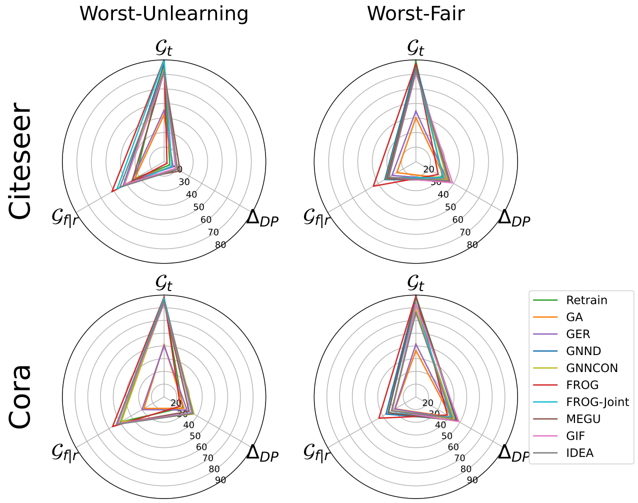

**Figure 5: Worst-case unlearning (left) and worst-case fairness (right) performance on CiteSeer and Cora.**

## 8 Results

### 8.1 RQ1: FROG Performance Under Uniform Removal

***Existing graph unlearning methods hurt fairness.*** Existing graph unlearning methods, though effective, detrimentally impact graph fairness post-unlearning. Results in Figure 3 show that GA, GNND, GNNCON have DP degradation of −0.089, −0.16, −0.21, −0.20, respectively. Notably, even Retrain compromises fairness by −0.064. GNND and GNNCON, though effective in unlearning with competitive scores on $\mathcal{G}_t$ and $\mathcal{G}_{f|r}$, suffer from the most significant degradation of fairness. This highlights that existing state-of-the-art graph unlearning models have overlooked graph fairness as an important factor to consider, which may hinder their application in fairness-concerned scenarios.

***FROG is effective in unlearning.*** In edge unlearning, FROG successfully distinguishes unlearned edges from retained edges measured by $\mathcal{G}_{f|r}$. As shown in Figure 3, when randomly removing 5% edges, FROG outperforms Retrain, GA, GER, GNND, and GNNCON by 10.0, 17.2, 19.7, 8.6, 5.9 absolute points, respectively. Under the node unlearning setting, as shown in Figure 4, FROG outperforms Retrain, GA, GER, GNND, and GNNCON by 5.7, 9.7, 13.0, 5.8, 3.9 absolute points, respectively, when deleting 5% of nodes. These results indicate that FROG exhibits more successful targeted knowledge removal of $\mathcal{G}_f$ than baselines. Meanwhile, FROG preserves model utility on downstream prediction tasks measured by $\mathcal{G}_t$, outperforming GA, GER, GNND, IDEA, MEGU by 32.2, 31.0, 6.0, 4.8,4.2,4.9 absolute points respectively, when deleting 5% of edges (Figure 3). FROG is even comparable to Retrain with a trivial gap of 2.2.

***FROG preserves fairness during unlearning.*** FROG consistently achieves the best fairness performance in terms of $\Delta_{EO}$, $\Delta_{SP}$, and $\Delta_{DP}$ on both edge and node unlearning tasks. For instance, in the edge unlearning task on the Cora dataset, FROG reduces $\Delta_{DP}$ by 21.6%, 39.5%, 35.4%, and 36.9% compared to GA, GNND, MEDU, and IDEA, respectively (Figure 3). In the node unlearning task, compared with IDEA, FROG reduces $\Delta_{EO}$ by 23.9% on Cora and 46.3% on CiteSeer, with comparable effectiveness (Figure 4). We attribute this superior performance over baselines to the fairness-aware design of the proposed method. Specifically, the optimization-based graph structure modification aims to find the optimal topology that results in both successful unlearning and minimum damage to fairness.

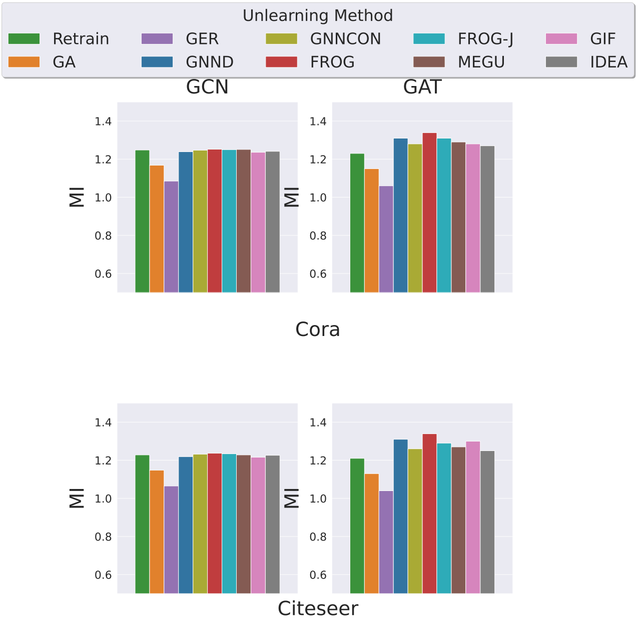

**Figure 6: Reported is the MI attack ratio. The left and right columns use GCN and GAT as backbones, respectively.**

***FROG effectively hides deleted information.*** Following [9], we delete 100 nodes and their associated edges from the training data and adopt Membership Inference attacks [23] on two datasets. Reported is the MI attack ratio. As shown in Figure 6, FROG outperforms other baseline methods, highlighting its effectiveness in hiding deleted information. Across two GNN architectures, we find that FROG achieves the highest MI ratio score of all baselines. Specifically, it outperforms GA, GER, GNNDelete, GIE, and IDEA by 0.089, 0.162, 0.035, 0.134, 0.086.

***Time Efficiency.*** We demonstrate that FROG is time-efficient compared to most unlearning baselines, as shown in Table 1. Specifically, FROG is **11.5×** faster than Retrain on CiteSeer and **1.25×** faster than IDEA and GNNCON. Although slightly slower than GNNDelete due to its structure learning component, FROG achieves significantly better predictive performance and fairness.

***Ablation Study.*** We examine the effect of the hyperparameter $\alpha$, which balances the unlearning and fairness objectives, as shown in Table 2. The results highlight the necessity of both $\mathcal{L}_{\text{un}}$ and $\mathcal{L}_{\text{fair}}$ for achieving a favorable AUROC–$\Delta_{DP}$ trade-off. As $\alpha$ increases, FROG places greater emphasis on reducing $\Delta_{DP}$. When $\alpha \leq 0.2$, edge unlearning performance remains stable; however, for $\alpha > 0.4$, AUROC drops sharply. To balance the trade-off, we set $\alpha = 0.2$ in our experiments.

**Table 1: Time spent when unlearning 100 edges on CiteSeer.**

| Method | Retrain | GA | GER | GNNDelete | GNNCON |
|---|---|---|---|---|---|
| Time (hr) | 0.63 | 0.31 | 0.26 | 0.025 | 0.05 |
| Method | MEGU | GIF | IDEA | FROG | FROG-Joint |
| Time (hr) | 0.04 | 0.10 | 0.06 | 0.04 | 0.03 |

**Table 2: Ablation study of effect on $\alpha$ on CiteSeer. Best performance is bold and second best is underlined.**

| $\alpha$ | $\lvert\mathcal{G}_f\rvert = 2.5\%$ | | | $\lvert\mathcal{G}_f\rvert = 5\%$ | | |
|---|---|---|---|---|---|---|
| | $\mathcal{G}_t(\uparrow)$ | $\mathcal{G}_{f\mid r}(\uparrow)$ | $\Delta_{DP}$ $(\downarrow)$ | $\mathcal{G}_t(\uparrow)$ | $\mathcal{G}_{f\mid r}(\uparrow)$ | $\Delta_{DP}$ $(\downarrow)$ |
| 0.0 | **90.7** | 79.0 | 32.9 | **90.1** | **71.3** | 37.5 |
| 0.1 | **90.7** | $\underline{79.1}$ | 29.5 | 89.9 | 70.8 | 29.2 |
| 0.2 | $\underline{90.6}$ | **79.6** | 25.8 | **90.1** | $\underline{71.0}$ | $\underline{25.5}$ |
| 0.4 | 89.1 | 76.2 | 23.7 | 89.9 | 70.4 | $\underline{25.5}$ |
| 0.6 | 87.2 | 75.7 | **23.5** | 89.2 | 70.1 | 25.7 |
| 0.8 | 90.3 | 72.1 | $\underline{26.7}$ | 88.9 | 69.5 | **24.9** |

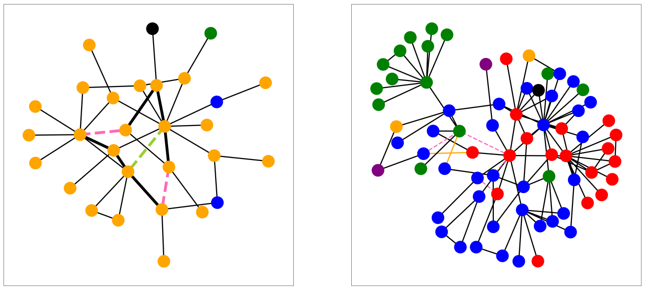

**Figure 7: Illustration of our method under worst-case evaluation. Pink dashed edges: edges to be removed. Green dashed edges: edges masked by FROG. Orange solid edges: edges to be added by FROG.**

## 8.2 RQ2: FROG Performance Under Worst-Case Scenarios

***Unlearning Performance.*** When $\mathcal{G}_f$ involves the set of edges that are the hardest to unlearn, we find that FROG is more effective to differentiate between $\mathcal{G}_f$ and $\mathcal{G}_r$ than baselines. This shows that in worst case, FROG still demonstrates more successful knowledge removal than existing graph unlearning methods, see Figure 5. We attribute this to the graph sparsification process, which simplifies hard-to-forget graph parts [24, 28, 29].

***Fairness Performance.*** Similarly when $\mathcal{G}_f$ involves the set of edges that are introduces the largest fairness degradation post-unlearning, FROG provides fairer representations than baselines, see Figure 5. The advantage on fairness inherits from the edge addition process, which injects new heterogeneous edges and dramatically mitigates network segregation. These results highlights the robustness of FROG to adversarial unlearning sets under extreme cases, making users more confident to apply FROG in real-world applications.

***Case Study.*** We present a case study in Figure 7. As a practical example for edge unlearning, we evaluate the performance of our algorithm in worst-case scenarios, as shown in Figure 7. In the left panel, the dashed pink edges represent hard-to-forget edges, identified using Equation 10. We observed that the two forgotten edges belong to loops (highlighted by bold edges) that continue facilitating message passing through their common neighbors even after removal. Consequently, these loops impede the unlearning of the target edges. To address this, our method proposes masking the dashed green edge, effectively breaking both loops simultaneously. In the right panel, the two worst-fairness edges obstruct message passing between different sensitive groups—one cluster dominated by green nodes and the other predominantly by blue and red nodes. To address this, the algorithm suggests adding two edges that connect the clusters without creating new loops.

## 9 Related Work

***Graph Unlearning.*** Machine unlearning on graphs [4] focuses on removing data influence from models. GraphEraser [4] approaches graph unlearning by dividing graphs into multiple shards and retrain a separate GNN model on each shard. However, this can be inefficient on large graphs and dramatically hurt link prediction performances. UtU formulates graph unlearning as removing redundant edges [29]. [7] develop a method to unlearn associations from multimodal graph-text data. CEU uses influence function for GNNs to achieve certified edge unlearning [16, 34]. However, how graph unlearning impacts the representation fairness of the retain graph remains unexplored.

***Graph Fairness.*** Numerous studies have explored fairness issues in graph learning. Fairwalk [26] introduced a random walk-based graph embedding method that adjusts transition probabilities based on nodes' sensitive attributes. Then in [22], they propose to use adversarial training on node embeddings to minimize the disparate parity. Then in [20], the focus shifted to dyadic fairness in link prediction, emphasizing that predictive relationships between instances should remain independent of sensitive attributes. Other works include fair collaborative filtering [37] in bipartite graphs and item recommendation tasks [3]. Only a few works examine the interplay of unlearning and fairness or bias at the same time [5, 6].

***Adversarial Unlearning.*** Several works try to stress-test unlearning methods under adversarial settings. On the method side, [13] investigates how adversarial training can help forget protected user attributes, such as demographic information. MUter [25] analyzes unlearning on adversarially trained models. On the evaluation side, [14] argues that unlearning should remove the generalization capability in addition to the data samples themselves. [12] studies adversarial unlearned data and proposes an approach to find challenging forget samples through bi-level optimization. [8] extends existing membership inference attacks to diverse data samples, even samples not in the training set, to check if a model has truly forgotten some knowledge. [5] uses counterfactual explanations to debias the machine unlearning algorithm in the classification task.

## 10 Conclusion

We presented FROG as the first graph unlearning method that not only effectively removes graph elements but also preserves the fairness of the retained graph. We formulated this problem as a bi-level optimization task that jointly optimizes the unlearned model and the underlying graph topology. Experiments on four datasets demonstrated that FROG can successfully unlearn information while preserving fairness. Furthermore, adversarial evaluations under challenging scenarios showed that FROG outperforms existing methods, achieving robust performance.

## Acknowledgments

This research was, in part, funded by the National Institutes of Health (NIH) Agreement No. 1OT2OD032581. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the NIH. This work was also partially supported by the following projects: SERICS (PE00000014) under the National Recovery and Resilience Plan funded by the European Union NextGenerationEU; HyperKG – Hybrid Prediction and Explanation with Knowledge Graphs (2022Y34XNM) funded by the Italian Ministry of University and Research under the PRIN 2022 program; GHOST – Protecting User Privacy from Community Detection in Social Networks (B83C24007070005) funded by Sapienza University of Rome - "Progetti di Ricerca Grandi."

## A Appendix

### A.1 Proofs

**Proof of Theorem 3.1:** For any pair of nodes coming from two different sensitive groups $v_i \in S_0, v_j \in S_1$, we have:

$$z_i - z_j = \mathbf{W}\left(\frac{1}{d_i+1}\sum_{v_p \in \mathcal{N}_i \cup v_i} X_p - \frac{1}{d_j+1}\sum_{v_q \in \mathcal{N}_j \cup v_j} X_q\right). \tag{14}$$

Based on the definition of $\rho$, among $|\mathcal{N}_i \cup v_i| = d_i + 1$ neighboring nodes of $v_i$, $\rho(d_i+1)$ of them come from the same sensitive feature distribution as $v_i$ while $(1-\rho)(d_i+1)$ of them come from the opposite feature distribution as $v_j$, then we have:

$$\begin{aligned}\frac{1}{d_i+1}\sum_{v_p \in \mathcal{N}_i \cup v_i} X_p &\sim \mathcal{N}(\rho\mu^0 + (1-\rho)\mu^1, \frac{1}{d_i+1}(\rho\Sigma^0 + (1-\rho)\Sigma^1));\\ \frac{1}{d_j+1}\sum_{v_q \in \mathcal{N}_j \cup v_j} X_q &\sim \mathcal{N}(\rho\mu^1 + (1-\rho)\mu^0, \frac{1}{d_j+1}(\rho\Sigma^1 + (1-\rho)\Sigma^0)).\end{aligned} \tag{15}$$

Consider $z_i - z_j$ follows a normal distribution, i.e., $z_i - z_j \sim \mathcal{N}(\mu, \Sigma)$, where $\mu = (2\rho - 1)\mathbf{W}(\mu^0 - \mu^1) = (2\rho - 1)\mathbf{W}\delta$. Let us denote $E_{i\in S_0}[z_i] = p$ and $E_{j\in S_1}[z_j] = q$. Thus:

$$\begin{aligned}\Delta_{DP} &= |E_{\substack{(i,j)\\ S_i=S_j}}[z_j \cdot z_i] - E_{\substack{(i,j)\\ S_i\neq S_j}}[z_i \cdot z_j]|\\ &= |p^T q - \Big(\frac{|S_0|^2}{|S_0|^2+|S_1|^2}p^T p + \frac{|S_1|^2}{|S_0|^2+|S_1|^2}\Big)q^T q| \qquad (16)\\ &\leq E||q-p||_2\Big(\frac{|S_0|^2}{|S_0|^2+|S_1|^2}p + \frac{|S_1|^2}{|S_0|^2+|S_1|^2}q\Big).\end{aligned}$$

Since $E|p-q| = \dot{(}2\rho - 1)\mathbf{W}\delta$, therefore it holds the following:

$$\Delta_{DP} \leq |K \cdot (2\rho - 1)\mathbf{W}\delta|.$$

**Proof of Theorem 5.1:** We first consider the case without structural modifications. In GNNDelete, $\mathbf{W}_D^L$ denotes the deletion matrix at layer $L$, and $\mathbf{h}_j^{L-1}$ represents the embedding of node $j$ from the previous layer. Following [9], We have $z_i = \sigma\Big(\sum_{j\in i\cup\mathcal{N}_i}\mathbf{W}\mathbf{h}_j^{L-1}\Big)$, After normalization, we have:

$$\langle z_i, z_j\rangle - \langle z_i', z_j'\rangle = \frac{1}{2}\|z_i' - z_j'\|^2 - \frac{1}{2}\|z_i - z_j\|^2. \tag{17}$$

Considering the deletion matrix and 17, we have:

$$\begin{aligned}\|z_i' - z_j'\| &= \|\sigma(\mathbf{W}_D^L z_i) - \sigma(\mathbf{W}_D^L z_j)\| & (18)\\ &\overset{\text{Lipschitz } \sigma}{\leq} \|\mathbf{W}_D^L z_i - \mathbf{W}_D^L z_j\| & (19)\\ &\overset{\text{Cauchy-Schwartz}}{\leq} \|\mathbf{W}_D^L\|\|z_i - z_j\| & (20)\end{aligned}$$

and applying that to Equation 17:

$$\langle z_i, z_j\rangle - \langle z_i', z_j'\rangle \leq \frac{1}{2}(\|\mathbf{W}_D^L\|^2 - 1)\|z_i - z_j\|^2. \tag{21}$$

Now we consider the structural modification. To forget edge $(i, j)$, let $C$ be their common neighbors. $C_i \in C$ and $C_j \in C$ are nodes with masked links to $i$ and $j$, and $C_{i\cap j} = C_i \cap C_j$. We can rewrite $z_i$ as:

$$z_i = \mathbf{W}\Big(\sum_{u\in C_i}\mathbf{h}_u^{L-1} + \sum_{v\in C_j}\mathbf{h}_v^{L-1} + \sum_{k\in C_{i\bigcap j}}\mathbf{h}_k^{L-1} + O_i\Big),$$

$$z_j = \mathbf{W}\Big(\sum_{u\in C_i}\mathbf{h}_u^{L-1} + \sum_{v\in C_j}\mathbf{h}_j^{L-1} + \sum_{k\in C_{i\bigcap j}}\mathbf{h}_k^{L-1} + O_j\Big),$$

Here, $O_i$ denotes the sum of representations of nodes that are neighbors of $i$ but not linked to $j$. After masking the redundant edges, we have:

$$z_i' = \sigma\Big(\mathbf{W}_D\mathbf{W}(\sum_{v\in C_j}\mathbf{h}_j^{L-1} + O_i)\Big), \quad z_j' = \sigma\Big(\mathbf{W}_D\mathbf{W}(\sum_{u\in C_i}\mathbf{h}_u^{L-1} + O_j)\Big)$$

$$\begin{aligned}\|z_i' - z_j'\| &= \|\sigma\Big(\mathbf{W}_D\mathbf{W}(\sum_{v\in C_j}\mathbf{h}_j^{L-1} + O_i)\Big) - \sigma\Big(\mathbf{W}_D\mathbf{W}(\sum_{u\in C_i}\mathbf{h}_u^{L-1} + O_j)\Big)\| & (22)\\ &\leq \|\sigma\Big(\mathbf{W}_D^L\mathbf{W}(O_i - O_j + \sum_{v\in C_j}\mathbf{h}_v^{L-1} - \sum_{u\in C_i}\mathbf{h}_u^{L-1})\Big)\| & (23)\\ &= \|\sigma\Big(\mathbf{W}_D^L(z_i - z_j) + \mathbf{W}_D^L\mathbf{W}(\sum_{v\in C_j}\mathbf{h}_v^{L-1} - \sum_{u\in C_i}\mathbf{h}_u^{L-1})\Big)\| & (24)\\ &\overset{\text{Lipschitz } \sigma}{\leq} \|\mathbf{W}_D\|\|z_i - z_j\| + \|\mathbf{W}^*\|\|\Delta\|, & (25)\end{aligned}$$

where $\Delta = \sum_{v\in C_j}\mathbf{h}_v^{L-1} - \sum_{u\in C_i}\mathbf{h}_u^{L-1}$. We also assume that $\sigma$ is a subadditive function, such as ReLU.

Combined with Equation 21, we could get the following:

$$\langle z_i, z_j\rangle - \langle z_i', z_j'\rangle \leq \Big(\frac{1}{2}\|\mathbf{W}_D^L\|^2 - 1\Big)\|z_i - z_j\|^2 + \|\mathbf{W}_D^L\mathbf{W}\|^2\|\Delta\|^2. \tag{26}$$

## GenAI Disclosure Statement

We used GPT-4 to identify and correct grammatical errors, typos, and to improve the overall writing quality. No AI tools were used at any other stage of this work to ensure full academic integrity.

## References

[1] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*.

[2] De Terwangne C. 2013. *The Right to be Forgotten and the Informational Autonomy in the Digital Environment*. Scientific analysis or review LB-NA-26434-EN-N. Luxembourg (Luxembourg). doi:10.2788/54562

[3] Abhijnan Chakraborty, Gourab K Patro, Niloy Ganguly, Krishna P Gummadi, and Patrick Loiseau. 2019. Equality of voice: Towards fair representation in crowdsourced top-k recommendations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 129–138.

[4] Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. 2022. Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*. 499–513.

[5] Ziheng Chen, Jia Wang, Jun Zhuang, Abbavaram Gowtham Reddy, Fabrizio Silvestri, Jin Huang, Kaushiki Nag, Kun Kuang, Xin Ning, and Gabriele Tolomei. 2024. Debiasing Machine Unlearning with Counterfactual Examples. *arXiv preprint arXiv:2404.15760* (2024).

[6] Jiali Cheng and Hadi Amiri. 2024. Mu-bench: A multitask multimodal benchmark for machine unlearning. *arXiv preprint arXiv:2406.14796* (2024).

[7] Jiali Cheng and Hadi Amiri. 2025. MultiDelete for Multimodal Machine Unlearning. In *Computer Vision – ECCV 2024*, Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (Eds.). Springer Nature Switzerland, Cham, 165–184.

[8] Jiali Cheng and Hadi Amiri. 2025. Tool Unlearning for Tool-Augmented LLMs. *arXiv preprint arXiv:2502.01083* (2025).

[9] Jiali Cheng, George Dasoulas, Huan He, Chirag Agarwal, and Marinka Zitnik. 2023. Gnndelete: A general strategy for unlearning in graph neural networks. *arXiv preprint arXiv:2302.13406* (2023).

[10] Zhihong Cui, Xiangguo Sun, Li Pan, Shijun Liu, and Guandong Xu. 2023. Event-based incremental recommendation via factors mixed Hawkes process. *Information Sciences* 639 (2023), 119007.

[11] Yushun Dong, Binchi Zhang, Zhenyu Lei, Na Zou, and Jundong Li. 2024. Idea: A flexible framework of certified unlearning for graph neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 621–630.

[12] Chongyu Fan, Jiancheng Liu, Alfred Hero, and Sijia Liu. 2024. Challenging forgets: Unveiling the worst-case forget sets in machine unlearning. In *European Conference on Computer Vision*. Springer, 278–297.

[13] Christian Ganhor, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. 2022. Unlearning Protected User Attributes in Recommendations with Adversarial Training. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Madrid, Spain) *(SIGIR '22)*. Association for Computing Machinery, New York, NY, USA, 2142–2147. doi:10.1145/3477495.3531820

[14] Shashwat Goel, Ameya Prabhu, Amartya Sanyal, Ser-Nam Lim, Philip Torr, and Ponnurangam Kumaraguru. 2022. Towards adversarial evaluations for inexact machine unlearning. *arXiv preprint arXiv:2201.06640* (2022).

[15] Aditya Golatkar, Alessandro Achille, and Stefano Soatto. 2020. Eternal Sunshine of the Spotless Net: Selective Forgetting in Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[16] Jin Huang, Zezhong Fan, Lalitesh Morishetti, Yuchan Guo, Kaushiki Nag, Hongshik Ahn, Ziheng Chen, and Gabriele Tolomei. 2025. Prompt-Tuning for Recommendation Unlearning. In *2025 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 859–863.

[17] Iris AM Huijben, Wouter Kool, Max B Paulus, and Ruud JG Van Sloun. 2022. A review of the gumbel-max trick and its extensions for discrete stochasticity in machine learning. *IEEE transactions on pattern analysis and machine intelligence* 45, 2 (2022), 1353–1371.

[18] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. 2021. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580* (2021).

[19] Oyku Deniz Kose and Yanning Shen. 2022. Fair contrastive learning on graphs. *IEEE Transactions on Signal and Information Processing over Networks* 8 (2022), 475–488.

[20] Peizhao Li, Yifei Wang, Han Zhao, Pengyu Hong, and Hongfu Liu. 2021. On dyadic fairness: Exploring and mitigating bias in graph connections. In *International Conference on Learning Representations*.

[21] Xunkai Li, Yulin Zhao, Zhengyu Wu, Wentao Zhang, Rong-Hua Li, and Guoren Wang. 2024. Towards Effective and General Graph Unlearning via Mutual Evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 13682–13690.

[22] Peiyuan Liao, Han Zhao, Keyulu Xu, Tommi S Jaakkola, Geoff Gordon, Stefanie Jegelka, and Ruslan Salakhutdinov. 2020. Graph adversarial networks: Protecting information against adversarial attacks. (2020).

[23] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. 2024. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, 4791–4809.

[24] Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, PRANAY SHARMA, Sijia Liu, et al. 2024. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems* 36 (2024).

[25] Junxu Liu, Mingsheng Xue, Jian Lou, Xiaoyu Zhang, Li Xiong, and Zhan Qin. 2023. MUter: Machine Unlearning on Adversarially Trained Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 4892–4902.

[26] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards fair graph embedding. (2019).

[27] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. Fairdrop: Biased edge dropout for enhancing fairness in graph representation learning. *IEEE Transactions on Artificial Intelligence* 3, 3 (2021), 344–354.

[28] Xiangguo Sun, Hong Cheng, Jia Li, Bo Liu, and Jihong Guan. 2023. All in one: Multi-task prompting for graph neural networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2120–2131.

[29] Jiajun Tan, Fei Sun, Ruichen Qiu, Du Su, and Huawei Shen. 2024. Unlink to Unlearn: Simplifying Edge Unlearning in GNNs. In *Companion Proceedings of the ACM Web Conference 2024* (Singapore, Singapore) *(WWW'24)*. Association for Computing Machinery, New York, NY, USA, 489–492. doi:10.1145/3589335.3651578

[30] Heinrich Von Stackelberg, Alan T Peacock, Erich Schneider, and TW Hutchison. 1953. The theory of the market economy. *Economica* 20, 80 (1953), 384.

[31] Yu Wang, Yuying Zhao, Yushun Dong, Huiyuan Chen, Jundong Li, and Tyler Derr. 2022. Improving fairness in graph neural networks via mitigating sensitive attribute leakage. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1938–1948.

[32] Zhepeng Wang, Runxue Bao, Yawen Wu, Guodong Liu, Lei Yang, Liang Zhan, Feng Zheng, Weiwen Jiang, and Yanfu Zhang. 2024. Self-guided knowledge-injected graph neural network for alzheimer's diseases. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 378–388.

[33] Jiancan Wu, Yi Yang, Yuchun Qian, Yongduo Sui, Xiang Wang, and Xiangnan He. 2023. Gif: A general graph unlearning strategy via influence function. In *Proceedings of the ACM Web Conference 2023*. 651–661.

[34] Kun Wu, Jie Shen, Yue Ning, Ting Wang, and Wendy Hui Wang. 2023. Certified Edge Unlearning for Graph Neural Networks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Long Beach, CA, USA) *(KDD '23)*. Association for Computing Machinery, New York, NY, USA, 2606–2617. doi:10.1145/3580305.3599271

[35] Yawen Wu, Dewen Zeng, Zhepeng Wang, Yiyu Shi, and Jingtong Hu. 2021. Federated contrastive learning for volumetric medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 367–377.

[36] Tzu-Hsuan Yang and Cheng-Te Li. 2023. When Contrastive Learning Meets Graph Unlearning: Graph Contrastive Unlearning for Link Prediction. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 6025–6032.

[37] Sirui Yao and Bert Huang. 2017. Beyond parity: Fairness objectives for collaborative filtering. *Advances in neural information processing systems* 30 (2017).

[38] Jiaxing Zhang, Dongsheng Luo, and Hua Wei. 2023. Mixupexplainer: Generalizing explanations for graph neural networks with data augmentation. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*. 3286–3296.

[39] Zizhou Zhang, Qinyan Shen, Zhuohuan Hu, Qianying Liu, and Huijie Shen. 2025. Credit Risk Analysis for SMEs Using Graph Neural Networks in Supply Chain. *arXiv preprint arXiv:2507.07854* (2025).