GraphGini: Fostering Individual and Group Fairness in Graph Neural Networks

Anonymous Author(s)

ABSTRACT

We address the growing apprehension that GNNs, in the absence of fairness constraints, might produce biased decisions that disproportionately affect underprivileged groups or individuals. Departing from previous work, we introduce for the first time a method for incorporating the Gini coefficient as a measure of fairness to be used within the GNN framework. Our proposal, GraphGini, works with the two different goals of individual and group fairness in a single system, while maintaining high prediction accuracy. GraphGini enforces individual fairness through learnable attention scores that help in aggregating more information through similar nodes. In addition, through Nash social welfare program, we ensure our solutions return pareto-optimal distribution of group fairness. ensures the maximum possible group fairness. Furthermore, GraphGini automatically balances all three optimization objectives (utility, individual, and group fairness) of the GNN and is free from any manual tuning of weight parameters. Extensive experimentation on real-world datasets showcases its efficacy in making significant improvements in individual fairness over state-of-the-art methods while maintaining utility and group equality.

KEYWORDS

Graph Neural Networks (GNNs), Fair GNN, Group Fairness, Individual Fairness, Gini Coefficient

1 INTRODUCTION

Graph Neural Networks (GNNs) are increasingly being adopted for various high-stake applications, including credit scoring for loan issuance [54], medical diagnosis [2], and recommendation engines [20, 26]. Recent studies have shown that GNNs may explicitly or implicitly inherit existing societal biases in the training data and in turn generate decisions that are socially unfair [11, 15]. As an example, Amazon disbanded an AI recruiting tool since it showed bias against women[12]. The bias infiltrated the AI model since most existing employers within AI were male, which was picked up as a desirable signal while analyzing the suitability of job applicants. It is therefore important to guard AI agents, including GNNs, from being influenced by sensitive features such as gender, race, religion, etc.

There are two primary approaches to address the issue of algorithmic bias: *individual fairness* and *group fairness* [40, 49]. Individual fairness involves masking the sensitive features and ensuring that individuals who are alike with respect to non-sensitive features receive similar treatment or outcomes from a system. For example, individual fairness would mean that two candidates with similar qualifications, experience, and skills should have similar chances of being selected, regardless of their personal characteristics such as gender, race, or age. The group fairness approach, on the other hand, looks within groups defined by sensitive features, e.g. female job applicants or male job applicants, and works to ensure that each such group has similar individual fairness characteristics to all other groups. For example, the variability in treatment between two highly qualified women candidates should be no more than the variability in treatment between two highly qualified male candidates. In general, having both properties is ideal.

1.1 Existing Works

The need to ensure fairness in ML models has been studied by several works [22, 43]. In this section, we focus on those works that are centered on fair GNNs [6] to achieve individual and/or group fairness. We note that the literature on fair GNNs include several other complementary areas spanning privacy-preserving fairness, adversarial attacks, sampling, etc. [6, 8, 14, 16, 17, 29, 30, 32, 33, 41, 42, 44–48, 52, 53, 55, 60, 66]

Group fairness: The objective in this paradigm is to achieve accurate prediction while also being independent of *protected* attributes that define groups. Rahman et al. [51] propose the idea of "equality of representation", which expands upon the concept of statistical parity for the node2vec model. Bose and Hamilton [3], and Dai and Wang [10] propose adversarial approaches to eliminate influence of sensitive attributes. He et al. [28] enables group fairness with respect to multiple non-binary sensitive attributes. Among other works, the strategies include balance-aware sampling [38], attention to mitigate group bias [36] and bias-dampening normalization [35].

Individual fairness: The most commonly used approach to improving individual fairness is the *Lipschitz constant* [19, 31, 37, 57]. Lipschitz constant is defined as follows. For any pair of individuals (u, v), let the input distance refer to the distance between u and v based on some domain-specific criteria (Eg., number of common friends, similarity on initial attributes, etc.), and the output distance be computed over their final GNN embeddings. Let $\epsilon_{u,v}$ be the ratio of output distance to input distance. The maximum $\epsilon_{u,v}$ across all pairs of nodes is defined to be the Lipschitz constant. Intuitively, a small Lipschitz constant implies that the distance in the embedding space is similar to the distance in the input space. Since the GNN prediction is a function of the embeddings, this transitively implies that similar individuals are likely to get similar predictions. A related, but alternative formulation is explored by Dong et al. [13], which operates in the ranking space instead of distances.

1.2 Limitations and Challenges

While the above referred works have made significant progress in the field of fair GNNs, certain challenges remain to be addressed.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

- **Grounding on social welfare literature:** The social welfare literature offers robust metrics for gauging both individual and group fairness. Well-known examples include the Gini Coefficient [24], Nash Social Welfare [4], EQ-1 equitability [21], EF1 envy-freeness [39], among others. However, current research on fair Graph Neural Networks (GNNs) largely neglects these established metrics. Instead, metrics like the Lipschitz constant have been independently developed and favored for GNN training, primarily due to their differentiability—an advantage not shared by metrics such as the Gini Coefficient. However, Lipschitz constant induces a major limitation; being a *max* operator, it is not robust to the presence of even a few outliers among the outcomes. In this work, we adopt the Gini coefficient. In addition to being an established social welfare metric, it operates on the entire spectrum of outcomes, providing a more comprehensive and robust measure of inequality. Appendix B contains an illustrative example showcasing a situation where the Lipschitz constant struggles to differentiate between different levels of inequality, whereas the Gini coefficient excels in this regard.
- **Ensuring Fairness on Both Group and Individual Levels:** Among existing studies, only Guide [57] integrates both group and individual fairness principles within a GNN framework. Balancing these two principles is intricate, as they may potentially conflict. Consider a university committed to both individual and group fairness in its admissions process. The university aims to admit students based on academic achievements and potential. However, historical disparities in educational resources may have led to lower average standardized test scores for a particular demographic group. Ensuring group fairness may entail adjusting the qualification bar for that group. *How do we design loss functions quantifying individual and group fairness in a harmonious manner? Furthermore, how should we avoid one dimension overpowering the others?* It's worth noting that some approaches aiming at achieving both individual and group fairness exist but operate in different directions. For instance, there are methods focusing on item ranking [23], post-processing inequality corrections [43], and measurement techniques [59]. However, our empirical evaluation reveals that when applied to graphs, these methods are less effective due to their lack of awareness of graph topology and limited flexibility in aligning outcomes with similarities across non-sensitive attributes.

1.3 Contributions

In this work, we design GraphGini, that addresses the above mentioned challenges. Specifically, our core contributions are as follows.

- **Problem Formulation Rooted in Social Welfare Literature:** We introduce the novel problem of developing a GNN that addresses both individual and group fairness. Notably, our formulation draws upon established metric of *Gini coefficient* from social welfare literature on algorithmic fairness to represent individual fairness. By doing so, we bridge the gap between fair GNN research and the broader community of algorithmic fairness, aligning our efforts with well-established principles in the field.
- **Algorithmic Innovation:** The conventional method of formulating a loss function as a weighted combination of quality and fairness metrics encounters challenges due to the non-differentiability of the Gini Index. To overcome this obstacle, we introduce a novel approach by establishing a differentiable upper bound on the Gini index. This is achieved by leveraging its association with to the average Lipschitz constraining scalar in terms of differentiable quadratic terms. Additionally, we unshackle the loss function from the manual selection of weights using GradNorm [7]. This empowers GraphGini to automatically adapt the weights based on the gradients observed for each optimization term. Consequently, it ensures a balanced optimization across all dimensions of fairness and prediction quality.
- **Empirical Evaluation:** We conduct a comprehensive benchmark of GraphGini using real-world datasets to demonstrate its superiority over existing fair GNN architectures. Our evaluation confirms that GraphGini outperforms state-of-the-art models in terms of quality while simultaneously maintaining equilibrium between individual and group fairness considerations.

2 PROBLEM FORMULATION

In this section, we introduce the preliminary concepts central to our work and formulate the problem of fostering individual and group fairness in GNNs.

2.1 Preliminaries

We use bold uppercase letters (e.g., $\mathbf{A}$) to denote matrices and $\mathbf{A}[\mathbf{i},:]$, $\mathbf{A}[:,\mathbf{j}]$ and $\mathbf{A}[\mathbf{i},\mathbf{j}]$ represent the i-th row, j-th column, and (i,j)-th entry of a matrix $\mathbf{A}$ respectively. For notational brevity, we use $\mathbf{z_i}$ to represent the row vector of a matrix, i.e., $\mathbf{z_i} = \mathbf{Z}[\mathbf{i},:]$). As convention, we use lowercase (e.g., n) and bold lowercase (e.g., $\mathbf{z}$) variable names to denote scalars and vectors, respectively. We use $\text{Tr}(\mathbf{A})$ to denote the trace of matrix $\mathbf{A}$. The l_1 and l_2-norm of a vector $\mathbf{z} \in \mathbb{R}^d$ are defined as $||\mathbf{z}||_1 = \sum_{\mathbf{i=1}}^{\mathbf{d}} |\mathbf{z_i}|$ and $||\mathbf{z}||_2 = \sqrt{\sum_{i=1}^{d} z_i^2}$ respectively. All notations are also summarized in Table F in Appendix.

Definition 1 (Graph). *A graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ has (i) n nodes ($|\mathcal{V}| = n$) (ii) a set of edges $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ and (iii) a feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ where each node is characterized by a d-dimensional feature vector.*

Several GNN architectures have been proposed in the literature [27, 34, 62]. We assume the following level of abstraction.

Definition 2 (Graph Neural Network (GNN) and Node Embeddings). *A graph neural network consumes a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ as input, and embeds every node into a c-dimensional feature space. We denote $\mathbf{Z} \in \mathbb{R}^{n \times c}$ to be the set of embeddings, where $\mathbf{z}_i$ denotes the embedding of $v_i \in \mathcal{V}$.*

Definition 3 (Sensitive attributes). *We assume that there is a sensitive attribute (e.g. gender) in the features of each individual, which partitions $\mathcal{V}$ into disjoint groups.*

Definition 4 (User similarity matrix $\mathbf{S}$ and Laplacian $\mathbf{L}$). *We assume, based on domain knowledge/application, there exists a similarity (or distance) function to capture the similarity between a pair of nodes $u_i, u_j \in \mathcal{V}$. $\mathbf{S}$ denotes the similarity matrix across all pairs of users. The Laplacian $\mathbf{L}$ of $\mathbf{S}$ is defined as $\mathbf{L} = \mathbf{D} - \mathbf{S}$ where, $\mathbf{D}$ is a diagonal matrix, with $\mathbf{D}[\mathbf{i},\mathbf{i}] = \sum_{j=1, j\neq i}^{n} \mathbf{S}[\mathbf{i},\mathbf{j}]$*

Without loss of generality, we assume the similarity value for a pair of nodes lies in the range $[0,1]$. The *distance* between nodes is defined to be the inverse of similarity, i.e., for two nodes $u_i, u_j \in \mathcal{V}$,

$d(u_i, u_j) = \frac{1}{S[i,j]+\delta}$, where δ is a small positive constant to avoid division by zero. Furthermore, we assume that the distance/similarity is symmetric and is computed after masking the sensitive attributes.

We next formulate the metrics for individual and group fairness. In economics theory, the *Lorenz* curve plots the cumulative share of total income held by the cumulative percentage of individuals ranked by their income, from the poorest to the richest [9, 24, 56]. The line at 45 degrees thus represents perfect equality of incomes. The *Gini coefficient* is the ratio of the area that lies between the *line of equality* and the Lorenz curve over the total area under the line of equality [24, 56] (See Fig. C in Appendix for an example). Mathematically,

$$Gini(\mathcal{S}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|s_i - s_j\right|}{2n\sum_{j=1}^{n} s_j} \tag{1}$$

where $\mathcal{S}$ is the set of individuals, s_i is the income of person i and $n = |\mathcal{S}|$. A lower Gini indicates fairer distribution, with 0 indicating perfect equality.

We adopt Gini to model individual fairness. Individual fairness demands that any two similar individuals should receive similar algorithmic outcomes [31]. In our setting, this implies that if two nodes (v_i and v_j) are similar (i.e., $S[v_i, v_j]$ is high), their embeddings ($\mathbf{z}_i$ and $\mathbf{z}_j$) should be similar as well. Inspired from [50], we encapsulate these requirements in the *weighted* Gini coefficient.

Definition 5 (Individual fairness). *The individual fairness of the node set $\mathcal{V}$ with embeddings $\mathbf{Z}$ and similarity matrix S is its weighted Gini coefficient, which is defined as follows.*

$$Gini(\mathcal{V}) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} S[i,j]\|\mathbf{z}_i - \mathbf{z}_j\|_1}{2n\sum_{i=1}^{n}\|\mathbf{z}_i\|_1} \tag{2}$$

The conventional definition of Gini (Eq. 1) assumes all individuals to be equal and hence expects an identical income distribution in line of equality. In our scenario, the similarity matrix S represents node similarity. Therefore, we formulate a weighted Gini coefficient where the difference between nodes is weighted according to their level of similarity. Under this formulation, significant disparities among dissimilar nodes do not significantly affect the Gini coefficient. Conversely, substantial disparities among similar nodes have a greater influence.

Optimizing individual fairness alone may cause group disparity [31]. Specifically, one group may have a considerably higher level of individual fairness than other groups. Hence, we employ the following group fairness satisfaction criteria to ensure equitable individual fairness across groups.

Definition 6 (Group Fairness). *Group Fairness is satisfied if the levels of individual fairness across all groups are equal. Mathematically, let $\{\mathcal{V}_1, \cdots, \mathcal{V}_m\}$ be the partition of node set $\mathcal{V}$ induced by the sensitive attribute. Group fairness demands $\forall i, j,\ Gini(\mathcal{V}_i) = Gini(\mathcal{V}_j)$.*

To convert group fairness satisfaction into an optimization problem, we introduce the notion of *group disparity of individual fairness*.

Definition 7 (Group Disparity). *Given a pair of groups $\mathcal{V}_g, \mathcal{V}_h$, the disparity among these groups is quantified as:*

$$GDIF(\mathcal{V}_g, \mathcal{V}_h) = \max\left\{\frac{Gini(\mathcal{V}_g)}{Gini(\mathcal{V}_h)}, \frac{Gini(\mathcal{V}_h)}{Gini(\mathcal{V}_g)}\right\} \tag{3}$$

More simply, $GDIF(\mathcal{V}_g, \mathcal{V}_h)\ \geq=\ 1$, with a a value of 1 indicating perfect satisfaction. The higher the value of disparity, the more is the deviation from the satisfaction criteria. Now, to generalize across m groups, *cumulative group disparity* measures the cumulative disparity across all pairs. Specifically,

$$C\text{-}GDIF(\{\mathcal{V}_1, \cdots, \mathcal{V}_m\}) = \sum_{\forall i,j\in[1,m],\ i\neq j} GDIF(\mathcal{V}_i, \mathcal{V}_j) \tag{4}$$

Empowered with the formal definitions of individual and group fairness, we finally state our problem objective.

Problem 1 (Fair GNN). *Given a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, a symmetric similarity matrix S for nodes in $\mathcal{V}$, and k disjoint groups differing in their sensitive attributes (i.e, $\cup_{i=1}^{k}\mathcal{V}_i$), our goal is to learn node embeddings $\mathbf{Z}$ such that:*

1. *Overall individual fairness level is maximized (Eq. 2);*
2. *Cumulative group disparity is minimized (Eq. 4).*
3. *The prediction quality on the node embeddings is maximized.*

3 GRAPHGINI: PROPOSED METHODOLOGY

The key challenge in learning fair node embeddings $\mathbf{Z}$ is to design a loss function that encapsulates all three objectives of Prob. 1. Designing this loss function requires us to navigate through three key challenges. **(1)** First, as we will establish in Proposition 1, Gini is non-differentiable and hence cannot be integrated directly into the loss function. **(2)** Second, since group disparity itself is a function of Gini, we require a differentiable proxy function to model group disparity as well. In addition, as we will show in Proposition 3, even if Gini is replaced with a differentiable proxy function, it remains non-differentiable. **(3)** Finally, we need a mechanism to automatically balance the three optimization objectives of the GNN without resorting to manual adjustments of weight parameters.

3.1 Optimizing Individual Fairness

Proposition 1. *The weighted Gini coefficient $Gini(\mathcal{V})$ defined in equation (2) is non-differentiable.*

Proof. For a pair of nodes v_i and v_j the L1 distance of their embeddings $\|\mathbf{z}_i - \mathbf{z}_j\|$ may be zero. The L1 norm is not differentiable around zero and so the sum of the L1 distances in the numerator makes Eq. 2 (as well as Eq. 1) non-smooth. This non-smoothness results in the Gini coefficient being non-differentiable at these points. □

To make the problem of optimizing individual fairness tractable, we design a *differentiable upper bound* on the Gini coefficient.

Proposition 2. *Given node embeddings $\mathbf{Z} \in \mathbb{R}^{n\times c}$ of graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ with node similarity matrix S and corresponding Laplacian $\mathbf{L}$ (Recall Def. 3), the weighted Gini coefficient $Gini(\mathcal{V})$ is upper bounded by the term $Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)$.*

Proof. From the study of norms we know that the L1 norm is lower bounded by the L2 norm, and that the L2 norm multiplied by the square root of the dimension of the space is an upper bound on the L1 norm. Using this fact we get:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} S[i,j]\|\mathbf{z}_i - \mathbf{z}_j\|_2 \leq \sum_{j=1}^{|\mathcal{E}|} S[i,j]\|\mathbf{z}_i - \mathbf{z}_j\|_1 \tag{5}$$

$$\leq \sqrt{c}\sum_{i=1}^{n}\sum_{j=1}^{n} S[i,j]\|\mathbf{z}_i - \mathbf{z}_j\|_2 \tag{6}$$

Now, substituting this inequality into the expression for $Gini(\mathcal{V})$ (Eq. 2), we have:

$$\frac{1}{2\sqrt{c}}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{S}[i,j]\|\mathbf{z}_i-\mathbf{z}_j\|_2 \le Gini(\mathcal{V})$$

$$\le \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{S}[i,j]\|\mathbf{z}_i-\mathbf{z}_j\|_2 = Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right) \quad (7)$$

□

Since the differentiable quadratic term $Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)$ serves as an upper bound for the weighted Gini coefficient $Gini(\mathcal{V})$, and is differentiable, it facilitates the minimization of $Gini(\mathcal{V})$ through the minimization of $Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)$. Hereon, we use the notation $\widehat{Gini}(\mathcal{V}) = Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right)$ to denote the upper bound on Gini with respect to node set $\mathcal{V}$ characterized by similarity matrix $\mathbf{S}$ and embeddings $\mathbf{Z}$.

3.2 Optimizing Group Fairness through Nash Social Welfare Program

Group fairness optimization, defined in Eq. 4, is a function of the Gini coefficient. Hence, Eq. 4 is non-differentiable as well. The natural relaxation to obtain a differentiable proxy function is therefore to replace $Gini(\mathcal{V}_g)$ with $\widehat{Gini}(\mathcal{V}_g) = Tr\left(\mathbf{Z}_g^T\mathbf{L}_g\mathbf{Z}_g\right)$ in Eq. 3. Here $\mathcal{V}_g \subseteq \mathcal{V}$ is a group of nodes, $\mathbf{Z}_g = \{\mathbf{z}_i \in \mathbf{Z} \mid v_i \in \mathcal{V}_g\}$ are the node embeddings of $\mathcal{V}_g$, and $\mathbf{L}_g$ is the Laplacian of the similarity matrix $\mathbf{S}_g : \mathcal{V}_g \times \mathcal{V}_g \to \mathbb{R}$. Unfortunately, even with this modification, optimization of group fairness remains non-differentiable.

Proposition 3. *Group disparity (Eq. 9), as defined below, is not differentiable.*

$$\widehat{GDIF}(\mathcal{V}_g,\mathcal{V}_h) = \max\left\{\frac{\widehat{Gini}(\mathcal{V}_g)}{\widehat{Gini}(\mathcal{V}_h)}, \frac{\widehat{Gini}(\mathcal{V}_h)}{\widehat{Gini}(\mathcal{V}_g)}\right\} \quad (8)$$

$$C\text{-}GDIF(\{\mathcal{V}_1,\cdots,\mathcal{V}_m\}) = \sum_{\forall i,j\in[1,m],\ i\neq j}\widehat{GDIF}(\mathcal{V}_i,\mathcal{V}_j) \quad (9)$$

Proof. Given a function of the form $max\{a,b\}$, it is differentiable everywhere except at the point where the two components are equal, i.e., $a=b$. However, at the point when $a=b$, which in our case happens at $\widehat{Gini}(\mathcal{V}_g) = \widehat{Gini}(\mathcal{V}_h)$ the max function exhibits a sharp corner or discontinuity. □

To circumvent Proposition 3, we employ *Nash social welfare program (NSWP)* optimization [5]. Specifically, we optimize the geometric mean, i.e.,

$$\text{Maximize } \left\{\widehat{Gini}(\mathcal{V}_g)\times\widehat{Gini}(\mathcal{V}_h)\right\} \quad (10)$$

$$\text{equivalently, Minimize } \left\{\frac{1}{\widehat{Gini}(\mathcal{V}_g)\times\widehat{Gini}(\mathcal{V}_h)}\right\} \quad (11)$$

To further influence the minimization problem in equitably distributing individual fairness across groups, we insert the term $\left(\widehat{Gini}(\mathcal{V}_g)-\widehat{Gini}(\mathcal{V}_h)\right)^2$ to the numerator, resulting in:

$$\text{Minimize } \left(\frac{(\widehat{Gini}(\mathcal{V}_g)-\widehat{Gini}(\mathcal{V}_h))^2}{\widehat{Gini}(\mathcal{V}_h)\widehat{Gini}(\mathcal{V}_g)}\right) \quad (12)$$

$$= -\left(\frac{\widehat{Gini}(\mathcal{V}_g)}{\widehat{Gini}(\mathcal{V}_h)}-1\right)\left(\frac{\widehat{Gini}(\mathcal{V}_h)}{\widehat{Gini}(\mathcal{V}_g)}-1\right) \quad (13)$$

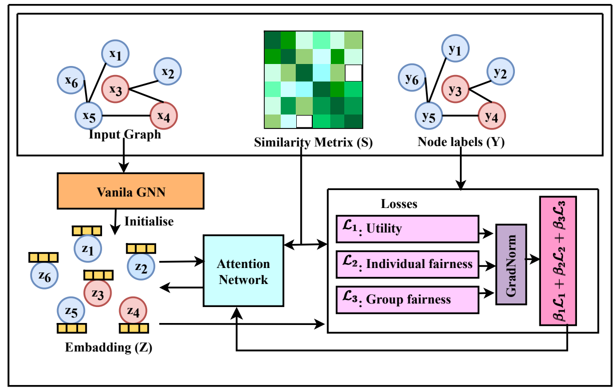


Figure 1: The figure illustrates the pipeline of GraphGini. The sequence of actions depicted in this figure is formally encapsulated in Alg. 1 and discussed in § 3.3.

Intuitively, group fairness is maximized when for any pair of groups $Gini(\mathcal{V}_g) = Gini(\mathcal{V}_h)$. On the other hand, if $Gini(\mathcal{V}_g) \gg Gini(\mathcal{V}_h)$ then Eq. 13 approaches ∞. We now establish that the solution obtained through minimizing Eq. 13 is *Pareto optimal*.

Proposition 4. *Minimizing Eq. 13 produces Pareto optimal solutions.*

Proof. Let the Gini upper bounds minimizing Eq. 13 be $Gini(\mathcal{V}_g) = \epsilon_g$ and $Gini(\mathcal{V}_h) = \epsilon_h$. The solution is Pareto optimal if we cannot reduce ϵ_h without increasing ϵ_g and vice-versa. We establish Pareto optimality through proof by contradiction. Suppose there exists as an assignment of node embeddings for which we get $Gini(\mathcal{V}_g) = \epsilon_g^*$, $Gini(\mathcal{V}_h) = \epsilon_h$ and $\epsilon_g^* > \epsilon_g$. Since Eq. 13 is minimized at ϵ_g and ϵ_h, this means that $-\left(\frac{\epsilon_g}{\epsilon_h}-1\right)\left(\frac{\epsilon_h}{\epsilon_g}-1\right) < -\left(\frac{\epsilon_g^*}{\epsilon_h}-1\right)\left(\frac{\epsilon_g}{\epsilon_h}-1\right)$, which is a contradiction. □

3.3 Learning Framework

GraphGini strives to achieve balance among three distinct goals:

(1) **Utility Loss:** We assume $\mathcal{L}_1$ denotes the loss term for the GNN prediction task. Naturally, we want to minimize $\mathcal{L}_1$.
(2) **Loss for individual fairness:** To maximize the overall individual fairness level, $Gini(\mathcal{V})$, we minimize:

$$\mathcal{L}_2 = -Tr\left(\mathbf{Z}^T\mathbf{L}\mathbf{Z}\right), \quad (14)$$

(3) **Loss for group fairness:** To maximize group fairness across a set of disjoint node partitions $\{V_1,\cdots,\mathcal{V}_m\}$, we minimize:

$$\mathcal{L}_3 = -\sum_{\forall i,j\in[1,m],\ i\neq j}\left(\frac{\widehat{Gini}(\mathcal{V}_i)}{\widehat{Gini}(\mathcal{V}_j)}-1\right)\left(\frac{\widehat{Gini}(\mathcal{V}_j)}{\widehat{Gini}(\mathcal{V}_i)}-1\right) \quad (15)$$

Incorporating all of the above loss terms, we obtain the overall loss function for GraphGini.

$$\mathcal{L} = \beta_1\mathcal{L}_1 + \beta_2\mathcal{L}_2 + \beta_3\mathcal{L}_3. \quad (16)$$

Here, β_is are tunable hyper-parameters that signify the importance of utility, individual fairness and group fairness, respectively.

The overall framework of GraphGini is presented in Fig. 1. Alg. 1 presents the pseudocode. First, embeddings $\mathbf{Z}$ for nodes are learned on the input GNN using only the utility loss. These initiation embeddings are now enhanced by injecting fairness. Specifically, we use a Graph Attention Network (GAT) [62] to calculate personalized attention weights for each node from its neighbourhood based on $\mathbf{S}$. The introduction of an attention mechanism facilitates the learning

Algorithm 1 GraphGini

```
Input: G = (V, E, X), S
Output: Fair Embeddings Z
Z ← Initialize GNN embeddings of V using utility Loss L_1
Initialize β_1 ← 1, β_2 ← 1, β_3 ← 1, t ← 0
Compute L_1, L_2 and L ← β_1 L_1 + β_2 L_2 + β_3 L_3.
while not converged in epoch t do
    for layer ℓ =[1 to K] do
        for each node v_i ∈ V do
            h_i^{ℓ+1} = σ(Σ_{j∈N_i} α_{i,j} W h_j^ℓ) // Message passing in GNN
    L ← β_1 L_1 + β_2 L_2 + β_3 L_3.
    Backpropagate using GradNorm loss corresponding to L (Eq. 20)
    Update W, α_{i,j}, β_1, β_2, β_3, and Z
    t ← t + 1
Return Z
```

of personalized aggregation weights to enhance embeddings based on the node features and pairwise similarity. The attention $\alpha_{i,j}$ from node v_i to node v_j in hidden layer ℓ is calculated by:

$$\alpha_{i,j} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i^\ell \| \mathbf{W}\mathbf{h}_j^\ell])\mathbf{S}[i,j])}{\sum_{j\in\mathcal{N}_i}\exp(\text{LeakyReLU}(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i^\ell \| \mathbf{W}\mathbf{h}_j^\ell])\mathbf{S}[i,j])}, \tag{17}$$

where $\mathbf{W} \in \mathbb{R}^{d\times h}$ is the weight matrix, $\mathbf{h}^\ell \in \mathbb{R}^h$ is the embedding for node i in layer ℓ, $\mathbf{a} \in \mathbb{R}^{2d}$ is the attention vector. The symbol $[.\|.]$ denotes concatenation of vectors and $\mathcal{N}_i$ represents the neighbourhood of node i. Using pairwise attention, the embeddings are aggregated as follows:

$$\mathbf{h}_i^{\ell+1} = \sigma(\Sigma_{j\in\mathcal{N}_i}\alpha_{i,j}\mathbf{W}\mathbf{h}_j^\ell), \tag{18}$$

The final embedding of node v_i is the embedding from the last layer K, i.e., $\mathbf{h}_i^K$. σ is an activation function. We assume the presence of a task-head, typically an MLP, that maps $\mathbf{h}_i^K$ to the output space.

3.4 Balanced Optimization with GradNorm

The optimisation process requires a delicate calibration of weighing factors, i.e., coefficients β_i's in our loss function (Eq. 16). Manual tuning, the predominant approach among existing techniques, is challenging as these weights not only determine the relative importance of fairness objectives compared to utility but also play a dual role in normalizing individual terms within the loss function. Specifically, the scale of $\mathcal{L}_1$, $\mathcal{L}_2$ and $\mathcal{L}_3$ can be vastly different. We perform *gradient normalization* to automatically learn the weights [7].

Let θ_t be the parameters of the GAT at epoch t and $\mathcal{L}_i(t),\ i \in \{1,2,3\}$ denotes the value of the loss term $\mathcal{L}_i$ in the t-th epoch. We initiate the learning process with initialized weights $\beta_i(0)$. In later rounds ($t > 0$), we treat each $\beta_i(t)$ as adjustable parameters aimed at minimizing a modified loss function, which we will develop next.

First, corresponding to each $\mathcal{L}_i(t)$, we compute the l_2 norm of the gradient.

$$G_i(t) = \|\Delta_{\theta_t}\beta_i(t)\mathcal{L}_i(t)\|_2 \tag{19}$$

The *loss ratio* in the t-th epoch is measured as $R_i(t) = \frac{\mathcal{L}_i(t)}{\mathcal{L}_i(0)}$, which reveals the inverse *training rate*, i.e., the lower the ratio, the higher is the training. Finally, instead of optimizing the original loss in $\mathcal{L}$ (Eq. 16), we minimize the loss below that operates on the gradient space of the individual loss terms. Specifically,

$$\mathcal{L}_{gradnorm}(t) = \sum_{i=1}^{3}\left\|\beta_i(t)G_i(t) - \bar{G}(t) \times \frac{R_i(t)}{\bar{R}(t)}\right\|_1 \tag{20}$$

$$\text{where, } \bar{G}(t) = \frac{1}{3}\sum_{i=1}^{3}G_i(t), \qquad \bar{R}(t) = \frac{1}{3}\sum_{i=1}^{3}R_i(t)$$

Table 1: Summary statistics of used datasets.

Name	# of nodes	# of node attributes	# of edges in A	# of edges in S	Sensitive Attribute
Credit	30,000	13	304,754	1, 687, 444	age
Income	14,821	14	100,483	1, 997, 641	race
Pokec-n	66,569	266	1,100,663	32, 837, 463	age

Intuitively, Eq. 20 strives to achieve similar training rates across all individual loss terms. Hence, it penalizes weighing factor $\beta_i(t)$ if the training rate is higher in $\mathcal{L}_i(t)$ than the average across all terms.

4 EXPERIMENTS

The objective in this section is to answer the following questions:

- **RQ1:** How well can GraphGini balance utility, individual fairness, and group fairness objectives compared to baselines?
- **RQ2:** How robust is GraphGini across GNN architectures?
- **RQ3:** Is GraphGini robust across similarity matrix variations?
- **RQ4:** Ablation study– What are the individual impacts of the various components utilized by GraphGini.

Each experiment is conducted five times, and the reported results consist of averages accompanied by standard deviations. Our experiments are performed on a machine with Intel(R) Core(TM) CPU @ 2.30GHz with 16GB RAM, RTX A4000 GPU having 16GB memory on Microsoft Windows 11 HSL. Our codebase is available at https://anonymous.4open.science/r/GraphGini-368F.

4.1 Datasets

Table 1 presents a summary of the real-world datasets used for benchmarking GraphGini.

Credit Dataset [65]: The graph contains 30,000 individuals, who are connected based on their payment activity. The class label whether an individual defaulted on a loan.

Income Dataset [57]: This is a similarity graph over 14,821 individuals sampled from the Adult Data Set [18]. The class label indicates whether an individual's annual income exceeds $50,000.

Pokec-n [61]: Pokec-n is a social network where the class label indicates the occupational domain of a user.

4.2 Empirical Framework

Metrics: The metric to assess node classification performance is evaluated through AUCROC and F1-score. To evaluate individual and group fairness we use *Gini* as well as the metrics used Guide [57]. These include overall individual (un)fairness (IF = $\text{Tr}(\mathbf{Z}^T\mathbf{L}\mathbf{Z})$) [31, 37] and Group disparity (GD). For two groups g and h, $GD = \max\{\epsilon_g/\epsilon_h, \epsilon_h/\epsilon_g\}$ where $\epsilon_g = \text{Tr}(\mathbf{Z}^T\mathbf{L}_g\mathbf{Z})$ and $\epsilon_h = \text{Tr}(\mathbf{Z}^T\mathbf{L}_h\mathbf{Z})$. We report the average GD across all pairs of groups. For group disparity, $GD = 1$ is the ideal case, with higher values indicating poorer performance.

Backbones GNNs: We evaluate on three distinct GNN backbones: GCN [34], GIN, [63], and Jumping Knowledge (JK) [64].

Baseline methods: We benchmark against the baselines below.

- Guide [57]: This method is the state-of-the-art, which minimizes the average Lipschitz constant for individual fairness and proposes a new group disparity measure based on the ratios of individual fairness among the groups.
- **FairGNN** [10]: This model leverages adversarial learning to ensure that GNNs achieve fair node classifications, adhering to group fairness criteria.

Table 2: Performance of benchmarked algorithms. Model indicates the debiasing algorithm and Vanilla represents no debiasing is performed. ↑ denotes the larger, the better; ↓ means the opposite. Best performances are in bold. Individual (un)fairness numbers are reported in thousands. All entries are averages and standard deviations.

Model	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)
					Credit				
		GCN			**GIN**			**JK**	
Vanilla	0.68±0.04	39.02±3.78	1.32±0.07	**0.71±0.00**	120.02±15.42	1.75±0.21	0.64±0.11	31.06±13.90	1.32±0.06
FairGNN	0.68±0.01	23.33±12.59	1.33±0.10	0.68±0.02	77.32±48.47	2.18±0.19	0.66±0.02	2.61±1.92	1.52±0.42
NIFTY	0.69±0.00	30.80±1.39	1.24±0.02	0.70±0.01	56.43±37.85	1.63±0.27	0.69±0.00	26.44±2.39	1.24±0.03
PFR	0.64±0.13	36.58±6.91	1.41±0.08	**0.71±0.01**	162.58±103.87	2.40±1.23	0.67±0.05	36.30±18.22	1.35±0.03
InFoRM	0.68±0.00	2.41±0.00	1.46±0.00	0.69±0.02	2.94±0.28	1.76±0.17	0.67±0.05	5.66±5.31	1.47±0.16
PostProcess	**0.70±0.00**	40.23±0.0	1.40±0.00	**0.71±0.00**	177.44±0.66	1.42±0.08	**0.70±0.00**	43.57±7.80	1.44±0.01
iFairNMTF	0.69±0.01	40.62±1.34	1.38±0.02	**0.71±0.01**	107.72±9.04	1.54±0.23	**0.70±0.00**	41.03±0.20	1.42±0.07
GNN GEI	0.69±0.00	40.21±1.32	1.35 ± 0.05	0.70±0.06	176.21±1.21	1.45 ± 0.23	0.68±0.00	43.16 ± 4.15	1.41±0.01
TF-GNN	0.69±0.00	12.80±0.00	1.45±0.00	**0.71±0.01**	18.42±0.42	1.29±0.11	**0.70±0.00**	13.20±0.03	1.43±0.01
Guide	0.68±0.00	1.93±0.11	**1.00±0.00**	0.68±0.00	2.43±0.02	**1.00±0.00**	0.68±0.00	2.34±0.11	**1.00±0.00**
GraphGini	0.68±0.00	**0.22±0.06**	**1.00±0.00**	0.68±0.00	**2.15±0.03**	**1.00±0.00**	0.68±0.00	**2.01±0.01**	**1.00±0.00**
					Income				
		GCN			**GIN**			**JK**	
Vanilla	0.77±0.00	369.11±0.03	1.29±0.00	**0.81±0.01**	2815.59±1047.33	1.87±0.48	**0.80±0.00**	488.73±166.83	1.18±0.16
FairGNN	0.76±0.00	249.73±87.53	1.17±0.04	0.79±0.00	1367.93±875.64	3.30±1.18	0.77±0.00	219.30±42.92	1.30±0.12
NIFTY	0.73±0.00	42.14±5.83	1.38±0.04	0.79±0.01	608.98±314.83	1.17±0.26	0.73±0.02	48.25±10.48	1.39±0.09
PFR	0.75±0.00	245.97±0.58	1.32±0.00	0.79±0.00	2202.64±445.24	2.36±1.17	0.73±0.13	327.57±155.49	1.12±0.23
InFoRM	**0.78±0.00**	195.61±0.01	1.36±0.00	0.80±0.01	308.45±13.92	1.62±0.30	0.79±0.00	192.58±12.87	1.35±0.11
PostProcess	0.77±0.00	367.62±0.00	1.28±0.00	0.80±0.00	420.78±128	2.5±0.01	0.79±0.00	520.23±20	1.27±0.01
iFairNMTF	0.77±0.00	358.20±0.32	1.28±0.01	0.80±0.00	2574.38±134.62	2.43±0.38	0.78±0.01	604.89±24.32	1.43±0.26
GNN GEI	0.77±0.00	357.23±5.04	1.47±0.01	0.79±0.00	2531.59±78.12	3.07±0.23	0.79±0.00	497.29±19.34	1.37±0.12
TF-GNN	0.76±0.00	25.65±0.00	1.85±0.00	0.80±0.01	310.20±1.20	1.28±0.01	0.78±0.01	205.08±03.25	1.48±0.01
Guide	0.73±0.01	33.19±10.17	**1.00±0.00**	0.74±0.02	83.88±20.29	**1.00±0.00**	0.74±0.01	42.49±21.93	**1.00±0.00**
GraphGini	0.73±0.09	**21.12±5.22**	**1.00±0.00**	0.74±0.00	**55.73± 9.12**	**1.00±0.00**	0.75±0.00	**31.23± 3.22**	**1.00±0.00**
					Pokec-n				
		GCN			**GIN**			**JK**	
Vanilla	**0.77±0.00**	951.72±37.28	6.90±0.12	**0.76±0.01**	4496.47±1535.62	8.35±1.24	**0.79±0.00**	1631.27±93.94	8.47±0.45
FairGNN	0.69±0.03	363.73±78.58	6.29±1.28	0.69±0.01	416.28±402.83	4.84±2.94	0.70±0.00	807.79±281.26	11.68±2.89
NIFTY	0.74±0.00	85.25±10.55	5.06±0.29	**0.76±0.01**	2777.36±346.29	9.28±0.28	0.73±0.01	477.31±165.68	8.20±1.33
PFR	0.53±0.00	98.25±9.44	15.84±0.03	0.60±0.01	628.27±85.89	6.20±0.79	0.68±0.00	729.77±74.62	15.66±5.47
InFoRM	**0.77±0.00**	230.45±6.13	6.62±0.10	0.75±0.01	271.65±30.63	6.83±1.34	0.78±0.01	315.27±25.21	6.80±0.54
PostProcess	**0.77±0.00**	872.12±82.23	5.93±0.27	0.75±0.00	4261.32±113.88	9.76±0.25	0.78±0.00	1721.42±83.91	10.22±0.45
iFairNMTF	0.76±0.00	781.29±98.45	7.23±0.11	0.75±0.00	3972.55±69.34	8.45±0.21	0.77±0.00	1602.52±92.73	9.37±0.10
GNN GEI	**0.77±0.00**	875.11±9.31	6.43±8.31	0.75±0.01	4383.26±319.56	7.29±0.87	0.78±0.00	1788.65±56.39	9.21±0.55
TF-GNN	0.74±0.00	245.48±11.43	9.28±0.10	0.75±0.00	268.32±21.82	9.31±1.22	0.76±0.00	418.31±54.26	10.20±1.45
Guide	0.73±0.02	55.05±30.87	1.11±0.03	0.74±0.01	120.65±17.33	1.12±0.03	0.75±0.02	83.09±18.70	1.13±0.02
GraphGini	0.74±0.00	**31.10±5.22**	**1.00± 0.00**	0.74±0.00	**85.10±6.29**	**1.00± 0.00**	0.78±0.10	**44.51±0.72**	**1.00± 0.00**

- **NIFTY** [1]: Addressing counterfactual fairness along with stability problem, NIFTY perturbs attributes and employs Lipschitz constants to normalize layer weights. Training incorporates contrastive learning techniques, and we adopt this model directly for various GNN backbone architectures.
- **PFR** [37]: PFR learns fair node embeddings as a pre-processing step, ensuring individual fairness in downstream tasks. The acquired embeddings serve as inputs for GNN backbones.
- **InFoRM** [31]: This model formulates an individual fairness loss within a graph framework based on the Lipschitz condition. We integrate the proposed individual fairness loss into the training process of GNN backbones.
- **PostProcess[43]:** Lohia et al. [43] proposes a post-processing based method to enhance individual and group fairness. The method employs a bias detector to assess disparity in outcomes, and when such biases are detected, it changes the model output to a different outcome. This algorithm is topology-agnostic.
- **GEI [59]**: GEI considers diversity of outcomes within a group, determined by sensitive attributes, as a measure of inequality. This implies the assumption that outcomes are independent of non-sensitive attributes within the group. In contrast, Gini allows weighting outcomes proportional to an input similarity measure (Eq. 1), leading to a more nuanced calculation of inequality based on similarity in non-sensitive attributes. To quantify this effect, we use GEI as the regularizer instead of Gini.
- **iFairNMTF [25]**: iFairNMTF is a fair clustering model that uses individual Fairness Nonnegative Matrix Tri-Factorization technique with contrastive fairness regularization to get balanced and cohesive clusters. We adapt iFairNMTF in our setting by plugging their fairness regularizer term with our GNN loss.
- **TF-GNN [58]**: TF-GNN presents an individual fair Graph Neural Networks (GNNs) tailored for the analysis of temporal financial transaction network data. In our specific context, we integrate TF-GNN into our framework by incorporating their fairness regularizer term into our GNN loss function.

Table 3: Gini coefficient for different clusters on Credit, Income, and Pokec-n datasets. The model indicates the algorithm, and Vanilla represents that no fairness mechanism has been used. Best performances are in bold. Individual (un)fairness numbers are reported in thousands. Here Van, GDE, and GGini corresponds to Vanilla, Guide and GraphGini respectively.

Model →	Van.	GDE	GGini	Van.	GDE	GGini	Van.	GDE	GGini
Credit									
	GCN			GIN			JK		
AUC(↑)	0.68	0.67	**0.69**	**0.71**	0.69	0.69	0.69	0.68	0.68
F1(↑)	0.71	0.72	**0.73**	0.72	0.73	**0.74**	0.70	0.71	**0.73**
IF(↓)	17.38	1.09	**1.01**	27.47	1.97	**1.78**	15.55	1.48	**1.42**
GD(↓)	1.35	**1.00**	**1.00**	1.87	**1.00**	**1.00**	1.24	**1.00**	**1.00**
Gini Cl 1 (↓)	0.08	**0.04**	0.04	0.11	0.08	**0.05**	0.11	0.04	**0.03**
Gini Cl 2 (↓)	0.10	0.08	**0.08**	0.13	0.04	**0.04**	0.14	**0.04**	0.05
Gini Cl 3 (↓)	0.27	**0.26**	0.26	0.29	0.06	**0.06**	0.25	**0.07**	0.08
Gini Cl 4 (↓)	0.20	0.19	**0.19**	0.17	0.13	**0.10**	0.19	0.08	**0.08**
Gini Cl 5 (↓)	0.09	0.07	**0.07**	0.11	**0.04**	0.05	0.13	0.04	**0.04**
Gini Cl 6 (↓)	0.08	0.05	**0.05**	0.11	**0.05**	0.06	0.13	0.03	**0.03**
Income									
	GCN			GIN			JK		
AUC(↑)	0.77	0.74	**0.78**	**0.81**	0.80	0.80	**0.80**	0.74	0.75
F1(↑)	0.78	**0.79**	0.78	0.80	**0.81**	**0.81**	0.79	**0.80**	**0.80**
IF(↓)	111.03	9.75	**6.85**	421.81	23.46	**22.31**	439.38	27.14	**24.39**
GD(↓)	1.23	**1.00**	**1.00**	1.16	**1.00**	**1.00**	1.29	**1.00**	**1.00**
Gini Cl 1 (↓)	0.34	0.19	**0.17**	0.43	**0.16**	0.17	0.40	**0.11**	0.11
Gini Cl 2 (↓)	0.28	0.18	**0.18**	0.38	0.02	0.17	0.34	**0.10**	0.15
Gini Cl 3 (↓)	0.32	0.15	**0.12**	0.56	0.15	**0.15**	0.47	**0.12**	0.16
Gini Cl 4 (↓)	0.21	0.08	**0.07**	0.21	0.15	**0.15**	0.24	0.12	**0.11**
Gini Cl 5 (↓)	0.37	**0.13**	0.13	0.60	0.25	**0.24**	0.49	0.39	**0.32**
Gini Cl 6 (↓)	0.22	**0.12**	0.13	0.24	**0.16**	**0.16**	0.25	0.16	**0.13**
Pokec-n									
	GCN			GIN			JK		
AUC(↑)	**0.77**	0.74	0.74	**0.76**	0.74	0.74	**0.79**	0.78	0.78
F1(↑)	0.75	0.77	**0.79**	0.76	0.77	**0.79**	0.76	**0.78**	**0.78**
IF(↓)	859.67	46.60	**26.16**	1589.50	96.90	**33.70**	1450.98	59.42	**44.51**
GD(↓)	2.55	**1.00**	**1.00**	3.50	**1.00**	**1.00**	3.07	**1.00**	**1.00**
Gini Cl 1	0.27	**0.10**	0.10	0.22	0.11	**0.10**	0.38	0.08	**0.07**
Gini Cl 2	0.17	0.09	**0.08**	0.13	**0.13**	0.14	0.34	0.09	**0.08**
Gini Cl 3	0.06	0.03	**0.02**	0.09	**0.06**	0.06	0.07	0.05	**0.04**
Gini Cl 4	0.27	**0.11**	0.13	0.15	0.10	**0.09**	0.35	0.07	**0.06**
Gini Cl 5	0.16	0.07	**0.06**	0.11	**0.11**	0.11	0.09	0.07	**0.07**
Gini Cl 6	0.27	**0.10**	0.11	0.21	0.10	**0.09**	0.08	0.05	**0.05**
Gini Cl 7	0.28	0.12	**0.10**	0.24	0.10	**0.10**	0.33	**0.10**	0.11
Gini Cl 8	0.27	0.16	**0.12**	0.29	0.12	**0.09**	0.33	0.10	**0.08**

The reproducibility details for baselines, hyper-parameter setting, and implementation specifications are given in the Appendix D.

Similarity matrix: In addition to the raw datasets, we also need a similarity matrix S (Def. 3). We evaluate on two different settings: **(1)** *topological similarity*, and **(2)** *attribute similarity*. To instantiate S for topological similarity, the (i, j)-th entry in S represents the cosine similarity between the i-th row and the j-th row of the adjacency matrix **A**. This is aligned with the similarity metric for evaluating fairness in existing works [31]. For the setting of attribute similarity, the (i, j)-th entry in S represents the cosine between the attributes of v_i and v_j after masking out the sensitive attributes.

4.3 RQ1 and RQ2: Efficacy of GraphGini and Robustness to Architectures

Table 2 presents a comprehensive evaluation of GraphGini against state-of-the-art baselines encompassing all three datasets, established metrics in the literature and three distinct GNN architectures. In this experiments, the similarity matrix is based on topological similarity. A clear trend emerges. While GraphGini suffers a minor decrease in AUCROC when compared to the vanilla backbone GNN, it comprehensively surpasses all baselines in individual and group fairness. Compared to Guide, which is the most recent and the only work to consider both individual and group fairness, GraphGini outperforms it across all metrics and datasets. Specifically, GraphGini is never worse in AUCROC, while always ensuring a higher level of individual and group fairness. In terms of numbers, for the Credit dataset, GraphGini improves individual fairness by 88%, 11%, and 14% as compared to Guide when embeddings are initialised by GCN, GIN and JK backbone architectures, respectively. Similarly, we observe significant improvement in individual fairness for the Income dataset by 36%, 33%, & 26% and 43%, 29 %, & 46 % for the Pokec-n dataset without hurting on utility and maintaining group fairness. Beyond Guide, we note that several of the baselines only optimize group fairness. Yet, GraphGini outperforms all of them in this metric while also optimizing individual fairness. Overall, this experiment surfaces the robustness of GraphGini to backbone GNN architectures and the efficacy of the proposed regularizers in ensuring fairness objectives when compared to existing baselines.

4.4 RQ3: Robustness to Similarity Matrix

In the next experiment, we use similarity matrix based on attribute similarity. In this case, the groups are created through k-means. The number of clusters are selected based on elbow plot (See Fig. D in Appendix for details). The primary objectives in this experiment are threefold. Does GraphGini continue to outperform Guide, the primary baseline, when similarity is on attributes? How well do these algorithms perform on the metric of Gini coefficient? How is the Gini coefficient distributed across groups (clusters)?

The findings are summarized in Table 3, mirroring the trends observed in Table 2. GraphGini consistently exhibits the best balance across all three metrics and outperforms Guide across all datasets and architectures on average. Additionally, we delve into the Gini coefficient analysis for each group (designated as ClX). As evident from Table 3, GraphGini achieves lower Gini coefficients across most clusters, underscoring the effectiveness of the regularizers.

4.5 RQ4: Ablation Study

Impact of Gradient Normalization: Table 4 presents the performance of GraphGini with all weights set to 1 (denoted as GraphGini WGN) against automatic tuning through gradient normalization. Gradient normalization imparts significant improvement in individual performance while achieving the same quality in group fairness and accuracy. Individual fairness (IF) benefits the most since the regularizer term corresponding to IF in our loss function is of the smallest magnitude. Hence, when set to equal weights, IF gets dominated by the other two terms in the loss. With gradient normalization, this issue is circumvented. Finally, it is noteworthy that even without gradient normalization, GraphGini-WGN outperforms Guide (refer to Table 2) across all datasets. This underscores that while gradient normalization contributes to improvement, it is not the sole reason for the superiority of GraphGini over Guide. More detailed insights into the evolution of the automatically-tuned weight parameters via gradient normalization are provided in App. E.

Table 4: Comparison of GraphGini with and without Gradient Normalization (GraphGini WGN).

Model	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)
Credit									
	GCN			GIN			JK		
GraphGini WGN	**0.68±0.00**	1.82±0.13	**1.00±0.00**	**0.68±0.00**	2.15±0.03	**1.00±0.00**	**0.68±0.00**	2.01±0.01	**1.00±0.00**
GraphGini	**0.68±0.00**	**0.22±0.06**	**1.00±0.00**	**0.68±0.00**	**1.92±0.09**	**1.00±0.00**	**0.68±0.00**	**1.88±0.02**	**1.00±0.00**
Income									
	GCN			GIN			JK		
GraphGini WGN	**0.73±0.09**	21.12±5.22	**1.00±0.00**	**0.74±0.00**	55.73± 9.12	**1.00±0.00**	**0.75±0.00**	31.23± 3.22	**1.00±0.00**
GraphGini	**0.73±0.09**	**20.50±3.10**	**1.00±0.00**	**0.74±0.00**	**49.03± 6.33**	**1.00±0.00**	**0.75±0.00**	**29.47± 4.01**	**1.00±0.00**
Pokec-n									
	GCN			GIN			JK		
GraphGini WGN	**0.74±0.00**	31.10±5.22	**1.00± 0.00**	**0.74±0.00**	85.10±6.29	**1.00± 0.00**	**0.78±0.10**	44.51±0.72	**1.00± 0.00**
GraphGini	**0.74±0.00**	**27.60±6.32**	**1.00± 0.00**	**0.74±0.00**	**81.37±9.87**	**1.00± 0.00**	**0.78±0.10**	**43.87±2.36**	**1.00± 0.00**

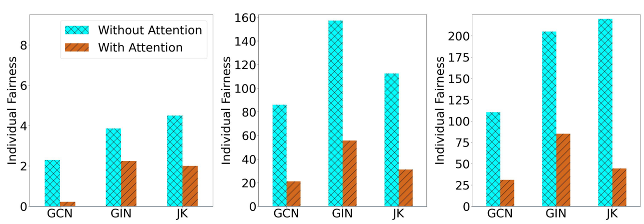
Table 5: Impact of fairness regularizers on performance.

(a) $\beta_2 = 0$, i.e., without individual fairness.

Model	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)
Credit									
	GCN			GIN			JK		
Guide	**0.68**	16.52	**1.00**	**0.68**	17.46	**1.00**	**0.68**	9.38	**1.00**
GraphGini	**0.68**	**13.77**	**1.00**	**0.68**	**13.35**	**1.00**	**0.68**	**9.28**	**1.00**
Income									
	GCN			GIN			JK		
Guide	**0.73**	26.07	**1.00**	**0.74**	348.69	**1.00**	0.74	73.04	1.00
GraphGini	**0.73**	**25.78**	**1.00**	**0.74**	**184.81**	**1.00**	**0.74**	**66.12**	**1.00**
Pokec-n									
	GCN			GIN			JK		
Guide	**0.74**	39.57	**1.00**	**0.74**	86.23	**1.00**	0.75	62.80	**1.00**
GraphGini	**0.74**	**33.42**	**1.00**	**0.74**	**38.00**	**1.00**	**0.76**	**41.67**	**1.00**

(b) $\beta_3 = 0$, i.e., without group fairness.

Model	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)	AUC(↑)	IF(↓)	GD(↓)
Credit									
	GCN			GIN			JK		
Guide	**0.68**	2.16	**1.52**	**0.68**	2.40	1.46	**0.68**	2.67	1.82
GraphGini	**0.68**	**2.11**	**1.52**	**0.68**	**2.37**	**1.45**	**0.68**	**2.56**	**1.52**
Income									
	GCN			GIN			JK		
Guide	**0.72**	25.38	**1.02**	0.74	152.35	1.23	**0.74**	439.27	1.14
GraphGini	**0.72**	**21.35**	1.09	**0.75**	**150.26**	**1.15**	**0.74**	**402.27**	**1.02**
Pokec-n									
	GCN			GIN			JK		
Guide	**0.74**	36.34	1.21	**0.74**	91.47	1.56	0.75	51.46	1.32
GraphGini	**0.74**	**22.14**	**1.11**	**0.74**	**87.27**	**1.41**	**0.76**	**46.9**	**1.22**



(a) Credit **(b)** Income **(c)** Pokec

Figure 2: Impact of attention on individual fairness.

Impact of attention: GraphGini integrates the inductive bias that message attention is influenced by the similarity between nodes (Eq. 17). How does this attention mechanism affect the outcomes? To investigate, we conducted an ablation study comparing results with and without attention. Our findings show that attention primarily improves individual fairness. Figure 2 illustrates these results on individual fairness. This observation is expected since attention is weighted based on similarity to neighbors, promoting nodes to prioritize similar neighbors in their embeddings. Consequently, the fundamental concept of individual fairness, which advocates for similar individuals receiving similar outcomes, is reinforced. **Impact of regularizers:** Next, we study the impact of the regularizers corresponding to individual and group fairness on the performance of GraphGini as well as Guide. To turn off a particular regularizer, we fix its weight to 0. Table 5 presents the results. Three key observations emerge. Firstly, as anticipated, both individual fairness and group fairness suffer when their respective regularizers are deactivated (compare the metrics of GraphGini and Guide in Table 5 with Table 2). Second, while group fairness remains unaffected from turning off individual fairness, the reverse is not true. This phenomenon occurs since group fairness (Eq. 3) is a function over individual fairness. Thus, even when individual fairness is not directly optimized, it gets indirect assistance from optimizing group fairness. Finally, GraphGini maintains its superiority over Guide, even with specific regularizers turned off.

5 CONCLUSION

In this work we have shown that how to combine the two key requirements of group fairness and individual fairness in a single GNN architecture GraphGini. The GraphGini achieves individual fairness by employing learnable attention scores that facilitate the aggregation of more information from similar nodes. Our major contribution is that we have used the well accepted Gini coefficient to define fairness, overcoming the difficulty posed by its non-dfferentiability. This particular way of using Gini is likely to have a major impact on future research since it provides a bridge between economics and machine learning. Our approach to group fairness incorporates the concept of Nash Social Welfare. Unlike existing state-of-the-art methods, the GraphGini automatically balances all three optimization objectives—utility, individual fairness, and group fairness—eliminating the need for manual tuning of weight parameters. To highlight the efficacy of our suggested framework, we conducted comprehensive experiments on real-world datasets. The findings show GraphGini significantly reduces individual unfairness while maintaining group disparity and utility performance after beating all state-of-the-art existing methods.

REFERENCES

[1] Chirag Agarwal, Himabindu Lakkaraju, and Marinka Zitnik. 2021. Towards a unified framework for fair and stable graph representation learning. In *Uncertainty in Artificial Intelligence*. PMLR, 2114–2124.

[2] David Ahmedt-Aristizabal, Mohammad Ali Armin, Simon Denman, Clinton Fookes, and Lars Petersson. 2021. Graph-based deep learning for medical diagnosis and analysis: past, present and future. *Sensors* 21, 14 (2021), 4758.

[3] Avishek Bose and William Hamilton. 2019. Compositional Fairness Constraints for Graph Embeddings. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 715–724.

[4] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. 2019. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)* 7, 3 (2019), 1–32.

[5] Hadi Charkhgard, Kimia Keshanian, Rasul Esmaeilbeigi, and Parisa Charkhgard. 2022. The magic of nash social welfare in optimization: Do not sum, just multiply! *The ANZIAM Journal* 64, 2 (2022), 119–134.

[6] April Chen, Ryan A Rossi, Namyong Park, Puja Trivedi, Yu Wang, Tong Yu, Sungchul Kim, Franck Dernoncourt, and Nesreen K Ahmed. 2023. Fairness-aware graph neural networks: A survey. *ACM Transactions on Knowledge Discovery from Data* (2023).

[7] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*. PMLR, 794–803.

[8] Sean Current, Yuntian He, Saket Gurukar, and Srinivasan Parthasarathy. 2022. Fairmod: Fair link prediction and recommendation via graph modification. *arXiv preprint arXiv:2201.11596* (2022).

[9] Camilo Dagum. 1980. The generation and distribution of income, the Lorenz curve and the Gini ratio. *Economie appliquée* 33, 2 (1980), 327–367.

[10] Enyan Dai and Suhang Wang. 2021. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.

[11] Enyan Dai, Tianxiang Zhao, Huaisheng Zhu, Junjie Xu, Zhimeng Guo, Hui Liu, Jiliang Tang, and Suhang Wang. 2022. A comprehensive survey on trustworthy graph neural networks: Privacy, robustness, fairness, and explainability. *arXiv preprint arXiv:2204.08570* (2022).

[12] Jeffrey Dastin. 2022. Amazon scraps secret AI recruiting tool that showed bias against women. In *Ethics of data and analytics*. Auerbach Publications, 296–299.

[13] Yushun Dong, Jian Kang, Hanghang Tong, and Jundong Li. 2021. Individual fairness for graph neural networks: A ranking based approach. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 300–310.

[14] Yushun Dong, Ninghao Liu, Brian Jalaian, and Jundong Li. 2022. Edits: Modeling and mitigating data bias for graph neural networks. In *Proceedings of the ACM web conference 2022*. 1259–1269.

[15] Yushun Dong, Jing Ma, Song Wang, Chen Chen, and Jundong Li. 2023. Fairness in graph mining: A survey. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[16] Yushun Dong, Song Wang, Jing Ma, Ninghao Liu, and Jundong Li. 2023. Interpreting unfairness in graph neural networks via training node attribution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 7441–7449.

[17] Yushun Dong, Binchi Zhang, Yiling Yuan, Na Zou, Qi Wang, and Jundong Li. 2023. Reliant: Fair knowledge distillation for graph neural networks. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*. SIAM, 154–162.

[18] Dheeru Dua and Casey Graff. 2017. UCI machine learning repository (2020). *URL http://archive. ics. uci. edu/ml* (2017).

[19] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference* (Cambridge, Massachusetts) *(ITCS '12)*. Association for Computing Machinery, New York, NY, USA, 214–226.

[20] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.

[21] Rupert Freeman, Sujoy Sikdar, Rohit Vaish, and Lirong Xia. 2019. Equitable allocations of indivisible goods. *arXiv preprint arXiv:1905.10656* (2019).

[22] David García-Soriano and Francesco Bonchi. 2021. Maxmin-fair ranking: individual fairness under group-fairness constraints. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 436–446.

[23] David García-Soriano and Francesco Bonchi. 2021. Maxmin-Fair Ranking: Individual Fairness under Group-Fairness Constraints *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 436–446.

[24] Joseph L Gastwirth. 1972. The estimation of the Lorenz curve and Gini index. *The review of economics and statistics* (1972), 306–316.

[25] Siamak Ghodsi, Seyed Amjad Seyedi, and Eirini Ntoutsi. 2024. Towards Cohesion-Fairness Harmony: Contrastive Regularization in Individual Fair Graph Clustering. *CoRR* abs/2402.10756 (2024). https://doi.org/10.48550/ARXIV.2402.10756 arXiv:2402.10756

[26] Lin Gong, Lu Lin, Weihao Song, and Hongning Wang. 2020. JNET: Learning User Representations via Joint Network Embedding and Topic Embedding. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. Association for Computing Machinery, 205–213.

[27] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[28] Yuntian He, Saket Gurukar, and Srinivasan Parthasarathy. 2023. Efficient Fair Graph Representation Learning Using a Multi-level Framework. In *Companion Proceedings of the ACM Web Conference 2023*. 298–301.

[29] Hussain Hussain, Meng Cao, Sandipan Sikdar, Denis Helic, Elisabeth Lex, Markus Strohmaier, and Roman Kern. 2022. Adversarial inter-group link injection degrades the fairness of graph neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 975–980.

[30] Zhimeng Jiang, Xiaotian Han, Chao Fan, Zirui Liu, Na Zou, Ali Mostafavi, and Xia Hu. 2022. Fmp: Toward fair graph message passing against topology bias. *arXiv preprint arXiv:2202.04187* (2022).

[31] Jian Kang, Jingrui He, Ross Maciejewski, and Hanghang Tong. 2020. InFoRM: Individual Fairness on Graph Mining. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*.

[32] Jian Kang, Yan Zhu, Yinglong Xia, Jiebo Luo, and Hanghang Tong. 2022. Rawlsgcn: Towards rawlsian difference principle on graph convolutional network. In *Proceedings of the ACM Web Conference 2022*. 1214–1225.

[33] Ahmad Khajehnejad, Moein Khajehnejad, Mahmoudreza Babaei, Krishna P Gummadi, Adrian Weller, and Baharan Mirzasoleiman. 2022. Crosswalk: Fairness-enhanced node representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11963–11970.

[34] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *CoRR* abs/1609.02907 (2016). arXiv:1609.02907 http://arxiv.org/abs/1609.02907

[35] Oyku Deniz Kose and Yanning Shen. 2022. Fast&fair: Training acceleration and bias mitigation for GNNs. *Transactions on Machine Learning Research* (2022).

[36] O Deniz Kose and Yanning Shen. 2023. FairGAT: Fairness-aware Graph Attention Networks. *arXiv preprint arXiv:2303.14591* (2023).

[37] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. 2020. Operationalizing Individual Fairness with Pairwise Fair Representations. *Proc. VLDB Endow.* 13, 4 (2020), 506–518.

[38] Xiao Lin, Jian Kang, Weilin Cong, and Hanghang Tong. 2023. BeMap: Balanced Message Passing for Fair Graph Neural Network. *arXiv preprint arXiv:2306.04107* (2023).

[39] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. 2004. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce*. 125–131.

[40] Haochen Liu, Yiqi Wang, Wenqi Fan, Xiaorui Liu, Yaxin Li, Shaili Jain, Yunhao Liu, Anil Jain, and Jiliang Tang. 2022. Trustworthy ai: A computational perspective. *ACM Transactions on Intelligent Systems and Technology* 14, 1 (2022), 1–59.

[41] Yazheng Liu, Xi Zhang, and Sihong Xie. 2022. Trade less Accuracy for Fairness and Trade-off Explanation for GNN. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 4681–4690.

[42] Zheyuan Liu, Chunhui Zhang, Yijun Tian, Erchi Zhang, Chao Huang, Yanfang Ye, and Chuxu Zhang. 2023. Fair graph representation learning via diverse mixture-of-experts. In *Proceedings of the ACM Web Conference 2023*. 28–38.

[43] Pranay K Lohia, Karthikeyan Natesan Ramamurthy, Manish Bhide, Diptikalyan Saha, Kush R Varshney, and Ruchir Puri. 2019. Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2847–2851.

[44] Donald Loveland, Jiayi Pan, Aaresh Farrokh Bhathena, and Yiyang Lu. 2022. Fairedit: Preserving fairness in graph neural networks through greedy graph editing. *arXiv preprint arXiv:2201.03681* (2022).

[45] Donald Loveland, Jiong Zhu, Mark Heimann, Ben Fish, Michael T Schaub, and Danai Koutra. 2022. On graph neural network fairness in the presence of heterophilous neighborhoods. *arXiv preprint arXiv:2207.04376* (2022).

[46] Jiaqi Ma, Junwei Deng, and Qiaozhu Mei. 2021. Subgroup generalization and fairness of graph neural networks. *Advances in Neural Information Processing Systems* (2021).

[47] Jing Ma, Ruocheng Guo, Mengting Wan, Longqi Yang, Aidong Zhang, and Jundong Li. 2022. Learning fair node representations with graph counterfactual fairness. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 695–703.

[48] Farzan Masrour, Tyler Wilson, Heng Yan, Pang-Ning Tan, and Abdol Esfahanian. 2020. Bursting the filter bubble: Fairness-aware network link prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 841–848.

[49] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *Comput. Surveys* 54, 6 (2021).

[50] Nuno Mota, Negar Mohammadi, Palash Dey, Krishna P Gummadi, and Abhijnan Chakraborty. 2021. Fair partitioning of public resources: Redrawing district

boundary to minimize spatial inequality in school funding. In *Proceedings of the Web Conference 2021*. 646–657.

[51] Tahleen Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. 2019. Fairwalk: Towards Fair Graph Embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization.

[52] Rebecca Salganik, Fernando Diaz, and Golnoosh Farnadi. 2022. Analyzing the Effect of Sampling in GNNs on Individual Fairness. *arXiv preprint arXiv:2209.03904* (2022).

[53] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. 2022. HM-EIICT: Fairness-aware link prediction in complex networks using community information. *Journal of Combinatorial Optimization* 44, 4 (2022), 2853–2870.

[54] Valentina Shumovskaia, Kirill Fedyanin, Ivan Sukharev, Dmitry Berestnev, and Maxim Panov. 2020. Linking bank clients using graph neural networks powered by rich transactional data: Extended abstract. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. 787–788.

[55] Uriel Singer and Kira Radinsky. 2022. Eqgnn: Equalized node opportunity in graphs. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 8333–8341.

[56] Thitithep Sitthiyot and Kanyarat Holasut. 2021. A simple method for estimating the Lorenz curve. *Humanities and Social Sciences Communications* 8, 1 (2021), 1–9.

[57] Weihao Song, Yushun Dong, Ninghao Liu, and Jundong Li. 2022. GUIDE: Group Equality Informed Individual Fairness in Graph Neural Networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA) *(KDD '22)*. Association for Computing Machinery, New York, NY, USA, 1625–1634.

[58] Zixing Song, Yuji Zhang, and Irwin King. 2023. Towards fair financial services for all: A temporal GNN approach for individual fairness on transaction networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2331–2341.

[59] Till Speicher, Hoda Heidari, Nina Grgic-Hlaca, Krishna P. Gummadi, Adish Singla, Adrian Weller, and Muhammad Bilal Zafar. 2018. A Unified Approach to Quantifying Algorithmic Unfairness: Measuring Individual and Group Unfairness via Inequality Indices. In *SIGKDD*. 2239–2248. https://doi.org/10.1145/3219819.3220046

[60] 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.

[61] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, Vol. 1.

[62] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *stat* 1050, 20 (2017), 10–48550.

[63] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2018. How Powerful are Graph Neural Networks? *CoRR* abs/1810.00826 (2018). arXiv:1810.00826 http://arxiv.org/abs/1810.00826

[64] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. 2018. Representation Learning on Graphs with Jumping Knowledge Networks. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). 5453–5462.

[65] I-Cheng Yeh and Che-hui Lien. 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications* 36, 2 (2009), 2473–2480.

[66] Huaisheng Zhu, Guoji Fu, Zhimeng Guo, Zhiwei Zhang, Teng Xiao, and Suhang Wang. 2023. Fairness-aware Message Passing for Graph Neural Networks. *arXiv preprint arXiv:2306.11132* (2023).

A NOTATIONS

Table F: Notations.

Notation	Description
G	input graph
$\mathcal{V}$	set of nodes in graph
$\mathcal{E}$	set of edges in graph
n	number of nodes in a graph
d	features dimension
$\mathcal{V}_h$	h^{th} group in graph
$\mathbf{A} \in \{0,1\}^{n\times n}$	adjacency matrix of graph G
$\mathbf{X} \in \mathbb{R}^{n\times d}$	node feature matrix of graph G
$\mathbf{Z} \in \mathbb{R}^{n\times c}$	output learning matrix of graph G with c number of features
$\mathbf{S} \in \mathbb{R}^{n\times n}$	pairwise similarity matrix of graph G
$\mathbf{L} \in \mathbb{R}^{n\times n}$	Laplacian similarity matrix
$Gini(\mathcal{V})$	Gini of node set $\mathcal{V}$

B ADVANTAGE OF GINI COEFFICIENT OVER LIPSCHITZ CONSTANT

Let us consider a population of 5 nodes, whose similarity matrix $\mathbf{S}$ is as follows.

	A	B	C	D	E
A	1	0.5	0.5	0.5	0.5
B		1	0.5	0.5	0.5
C			1	0.5	0.5
D				1	0.5
E					1

We only show the upper triangle due to symmetry of the similarity function. The distance between a pair of nodes i, j is defined as its inverse, i.e., $d_{ij} = \frac{1}{\mathbf{S}[i,j]}$. Thus, in our case, all pairs of distinct nodes have a distance of 2.

Corresponding to this similarity/distance matrix, let us consider two GNNs that produce the embeddings in Table G.

Table G: Embeddings of two GNNs.

Node	Embeddings from GNN-1	Embeddings from GNN-2
A	[10, 10, 10, 10]	[10, 10, 10, 10]
B	[1, 0, 0, 0]	[1, 0, 0, 0]
C	[0, 1, 0, 0]	[0, 2, 0, 0]
D	[0, 0, 1, 0]	[0, 0, 2, 0]
E	[0, 0, 0, 1]	[0, 0, 0, 2]

As can be seen, while GNN-1 preserves an identical distance of 2 in the output space among all nodes except those involving A, the inequality is higher in GNN-2. Despite this, the Lipschitz constant for both GNN-1 and GNN-2 are dominated by the (A, B) pair leading to an identical value of 19.5. Gini coefficient (Eq. 2), being an operator over the entire output spectrum, yields values of 0.377 and 0.74 for GNN-1 and GNN-2 respectively. This difference in Gini coefficients accurately reflects the higher inequality present in GNN-2's outcomes, where a higher Gini coefficient signifies greater inequality. From a social welfare perspective, there will inevitably be individuals with significant wealth, making it impractical to eliminate them entirely. The Gini coefficient acknowledges this reality by aiming to quantify similarity-weighted inequality across all pairs. On the other hand, Lipschitz, being based on a maximum criterion, lacks robustness when dealing with even a small number of outliers.

C LORENZ CURVE

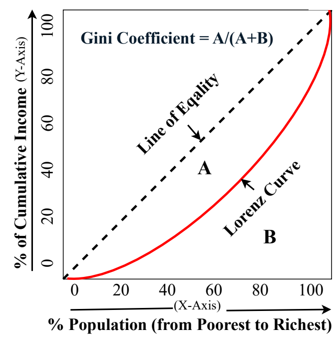


Figure C: Lorenz curve: It is the plot of the proportion of the total income of the population (on the y-axis) cumulatively earned by the bottom x (on the x-axis) of the population. The line at 45 degrees thus represents perfect equality of incomes. The further away the Lorenz curve is from the line of perfect equality, the greater the inequality. The Gini coefficient is the area ratio between the line of equality and the Lorenz curve over the total area under the line of quality.

D IMPLEMENTATION DETAILS FOR REPRODUCIBILITY

In this section, we state the implementation specifics of GraphGini and other baseline methods presented in Section 4. For all three datasets, we employ a random node shuffling approach and designate 25% of the labeled nodes for validation and an additional 25% for testing purposes. The training set sizes are set at 6,000 labeled nodes (25%) for the Credit dataset, 3,000 labeled nodes (20%) for the Income dataset, and 4,398 labeled nodes (6%) for Pokec-n. For the Pokec-n dataset, friendship linkages serve as edges, while for the remaining datasets, edges are not predefined, necessitating their construction based on feature similarity. More precisely, we establish a connection for any given pair of nodes if the Euclidean distances between their features surpass a predetermined threshold. The fine-tuned hyper-parameters used to train baselines are given in Table H. β_1 is 1 for all baselines.

E IMPACT OF GRADIENT NORMALIZATION

In Fig. E, we plot the trajectory of the weights against training epochs. In the credit dataset, initially, utility loss is given higher weightage than group fairness loss, but after certain iterations, group fairness loss weightage overcomes utility loss weightage. Meanwhile, the utility loss is always given the higher weightage in the other two datasets. These behaviors indicate the sensitivity of

Table H: Baseline hyper-parameters. – indicates parameter not used to train the model.

Datasets →	Credit		Income|		Pokec-n	
Model ↓	β_2	β_3	β_2	β_3	β_2	β_2
FairGNN	4	1000	4	10	4	100
NIFTY	-	-	-	-	-	-
PFR	-	-	-	-	-	-
PostProcess	-	-	-	-	-	-
iFairNMTF	1e-7	-	1e-7	-	1e-7	-
GNN GEI	1	-	1	-	1	-
TF-GNN	1e-6	-	1e-7	-	1e-7	-
InFoRM	5e-6	-	1e-7	-	1e-7	-
GUIDE	5e-6	1	1e-7	0.25	2.5e-7	0.05

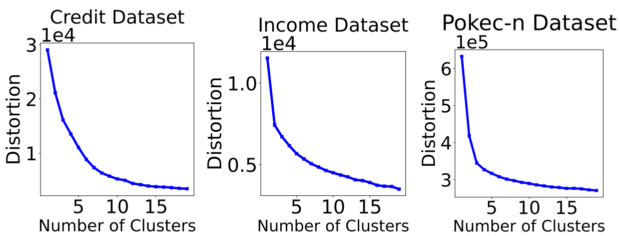
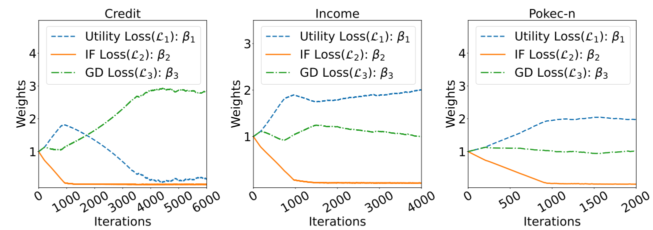


Figure D: The Elbow plots show the optimal number of clusters in each dataset on K-means clustering.



(a) Credit **(b)** Income **(c)** Pokec

Figure E: Adaptive Weights during training for each loss in GraphGini through GradNorm with GCN backbone.

β_i. With the increase in the number of iterations/epochs, initially, β_is' are changing at different rates, but after a certain iteration, the values of β_is' get stabilized, indicating similar training rates across all three losses.