[Re] Learning Fair Graph Representations via Automated Data Augmentations

Anonymous authors
Paper under double-blind review

Abstract

Due to the expanding reproducibility crisis in machine learning, there is a pressing need for careful evaluation of research findings. We evaluate the reproducibility of the paper "Learning Fair Graph Representations via Automated Data Augmentations" by Ling et al. (2023). Our objective is to assess the authors' three major claims: (1) fair augmentations improve fairness while retaining similar accuracy compared to other fairness methods, (2) augmenting both edges and node features performs better than augmenting only one of the two, and (3) learned augmentations reduce node-wise sensitive homophily and correlation between node features and the sensitive attribute. To achieve this, we (a) repeat all experiments from the original paper; (b) implement a new multi-run evaluation protocol with different random seeds; (c) perform extensive ablations of components of Graphair, and (d) investigate the generalizability of the method to other graph neural network (GNN) architectures, and graphs with varying homophily. Our results indicate that claims (1), (2), and (3) are only partially reproducible, achieving comparable performance on a maximum of two out of the three original datasets. Notably, the specific dataset(s) where the claims fail to hold varies between experiments. Additionally, we found the method generalizes to other GNN architectures, but does not generalize to graphs with varying homophily, failing for unbalanced homophily settings. Finally, throughout our experiments, we uncover a lack of stability in the Graphair framework.

1 Introduction

The reproducibility crisis identified in machine learning (Kapoor & Narayanan, 2022) highlights the critical necessity for replication and validation in research. Our investigation focuses on reproducing and extending the work by Ling et al., 2023, which proposed a fair learning method for Graph Neural Networks (GNNs; Kipf & Welling, 2017).

GNNs have shown considerable performance (Gao & Ji, 2019; Liu et al., 2021a;b; Yuan et al., 2021) across multiple domains (Hamaguchi et al., 2017; Liu et al., 2022; Han et al., 2022; Hamilton et al., 2017). Despite this success, GNNs have been shown to carry over or amplify bias present in training data (Dai & Wang, 2021; Mehrabi et al., 2021; Olteanu et al., 2019). Specifically, the output of neural networks can be substantially correlated with sensitive features such as ethnicity, nationality, and gender, even though this behavior might not have been intended (Dai & Wang, 2021). Such biases can be present in graph data, hidden in both the way nodes are connected and their features (Mehrabi et al., 2021; Olteanu et al., 2019). Moreover, removing sensitive features does not always lead to a fairer dataset, as the sensitive information can be retrieved from other features or their combinations (Sweeney, 2002).

To tackle this issue, previous approaches for fair graph models have employed various pre-processing techniques (i.e. augmenting the data to remove bias before training the actual network). Some augmentation-based methods used handcrafted heuristics for changing the training data (Spinelli et al., 2021; Kose & Shen, 2022). Building upon this foundation, Ling et al. (2023) propose *Graphair*, a framework focused on learning data augmentations that remove bias from a dataset. The authors claim that this method considerably improves on common group fairness metrics across multiple datasets compared to other fair graph

training schemes while achieving similar performance in terms of accuracy. We reproduce the experiments by Ling et al. (2023) to verify their main claims, as well as extend the tests of the method to examine its generalizability and robustness. More specifically, our contributions are summarized below:

1. We repeat all experiments from the original paper by Ling et al. (2023).
2. We alter the evaluation procedure to retrain the entire method five times with different random seeds, thereby achieving a more robust assessment of the method's stability and performance.
3. We assess how well the method generalizes to different architectures.
4. We conduct further ablation studies to investigate how the individual components of the Graphair framework contribute to improving fairness.
5. We run additional experiments with synthetic datasets to assess the stability and performance of Graphair for datasets with different homophily.
6. We improve the readability and reproducibility of the given code.

2 Scope of Reproducibility

Ling et al. (2023) propose a technique that automatically learns augmentations that change both the adjacency matrix and the node embeddings of a graph by incorporating adversarial and contrastive learning approaches. We identify the following claims from the paper:

Claim 1: Learned augmentations **improve group fairness metrics** while retaining **similar accuracy** compared to other fair graph-learning methods, on datasets containing sensitive attributes.

Claim 2: Augmenting both edges and node features achieves higher accuracy and better fairness scores than just augmenting one of them.

Claim 3: Learned augmentations **reduce node-wise sensitive homophily and correlation** between node features and the sensitive attribute.

3 Methodology

We use the code provided by the authors[1], with various improvements, such as setting random seeds for reproducibility, putting all the hyperparameters into a single file to make further replication and experimentation easier, adding missing functionality that was described in the paper, but did not appear in the codebase, as well as bug fixes. It is important to mention that the code provided by the authors is not the original code they used for running the experiments. Therefore, we attempt to make the code as similar to the settings described in the paper as possible. We also extend the authors' work with further experiments, which we describe in section 3.4. The code for the reproduction and extensions is available online[2].

3.1 Graphair framework overview

Given a graph $G = \{A, X, S\}$ with an adjacency matrix A, node features X, and the sensitive feature S, the Graphair method employs an augmentation module g to produce fair augmented data, an adversary k to ensure fairness, and a GNN-based encoder f to produce node-embeddings for downstream tasks. The three components are described in detail below.

[1] `https://github.com/divelab/DIG`
[2] `https://anonymous.4open.science/r/FACT-B421`

3.1.1 Augmentation Module

The augmentation module g first uses a Graph Convolutional Network (GCN; Kipf & Welling, 2017) encoder g_{enc} to obtain node embeddings $Z = g_{enc}(A, X)$. The embedding Z is used to create a new graph $G' = \{A', X', S\}$

$$Z_A = \text{MLP}_A(Z), \quad \widetilde{A'} = \sigma\left(Z_A Z_A^T\right), \quad A'_{ij} \sim \text{Bernoulli}\left(\widetilde{A'}_{ij}\right) \text{ for } i, j = 1, \cdots, n$$

$$Z_X = \text{MLP}_X(Z), \quad \widetilde{M} = \sigma\left(Z_X\right), \quad M_{ij} \sim \text{Bernoulli}\left(\widetilde{M}_{ij}\right) \text{ for } i, j = 1, \cdots, n$$

$$X' = M \odot X$$

where MLP_A and MLP_X are two Multi-Layer Perceptrons (MLPs), σ is the sigmoid function, and $\odot$ is the Hadamard product. MLP_A and MLP_X both have two linear layers, separated by a ReLU activation function. To make the sampling process differentiable, the Gumble-Softmax reparameterization trick is employed (Jang et al., 2016).

3.1.2 Adversarial Training

To ensure fairness, Graphair uses adversarial training: a GCN with an MLP head is defined as the adversary $k : (A', X') \to \hat{S} \in [0, 1]^n$, which predicts the sensitive attribute S from the augmented adjacency matrix A' and masked node embeddings X'. The adversarial module k is trained together with the augmentation module g in an adversarial fashion, utilizing an adversarial loss L_{adv}. While the adversarial module tries to predict the sensitive feature from A' and X', the augmentation module is trained so that A' and X' do not contain bias, making it difficult for the adversary to recover the sensitive attribute. The adversarial training procedure can formally be described as the following optimization problem:

$$\min_g \max_k L_{\text{adv}} = \min_g \max_k -L_{\text{BCE}}\left(S_i, \hat{S}_i\right)$$

where L_{BCE} denotes binary cross-entropy, S_i denotes the sensitive feature and $\hat{S}_i$ denotes the prediction of the sensitive feature by the adversary.

3.1.3 Contrastive Learning and Reconstruction Loss

Additionally, to avoid the loss of information and converging to a trivial solution, a contrastive learning objective L_{con} is used, which maximizes agreement between the representations of the original graph and the augmented one. These representations are denoted h and h' respectively and are learned by the GNN-based encoder f:

$$L_{\text{con}} = \frac{1}{2n} \sum_{i=1}^{n} \left[l\left(h_i, h'_i\right) + l\left(h'_i, h_i\right)\right]$$

where $l\left(a, b\right)$ denotes the contrastive learning objective defined by Zhu et al. (2020).

Finally, a reconstruction loss term L_{rec} is added to avoid large deviations of the augmented graph from the original graph.

$$L_{\text{rec}} = L_{\text{BCE}}\left(A, \widetilde{A'}\right) + \lambda L_{\text{MSE}}\left(X, X'\right)$$

where L_{BCE} and L_{MSE} denote binary cross entropy and mean squared error, $\widetilde{A'}$ refers to the edge probability matrix obtained from the augmentation module, and λ is a hyperparameter.

The overall training process can be described by the following min-max optimization problem:

$$\min_{f,g} \max_k L = \min_{f,g} \max_k \alpha L_{\text{adv}} + \beta L_{\text{con}} + \gamma L_{\text{rec}}$$

where α, β and γ are hyperparameters. This scheme defines a self-supervised training regime that aims to produce a fair GNN-based encoder f as the end product for being used in downstream tasks. Appendix 5 shows a visual overview of the framework.

3.2 Evaluation

To evaluate the Graphair framework, the embeddings produced by the trained encoder are used for a downstream task. For this purpose, a simple MLP classifier is trained on the embeddings to predict a binary target attribute Y. The classifier's performance is measured by its test accuracy, as well as two group fairness metrics: Demographic Parity ΔDP and Equal Opportunity ΔEO (Louizos et al., 2015; Beutel et al., 2017), which are defined as follows:

$$\Delta\text{DP} = |P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)|$$

$$\Delta\text{EO} = |P(\hat{Y} = 1|S = 0, Y = 1) - P(\hat{Y} = 1|S = 1, Y = 1)|$$

where $\hat{Y}$ is the prediction of the classifier, Y is the true target label, and S is the sensitive feature. The MLP network consists of 2 linear layers with a ReLU activation in between. The classifier is trained 5 times and the average metrics are reported.

3.3 Datasets

For the replication, we use the same three datasets as the original paper: NBA, Pokec-z, and Pokec-n (Dai & Wang, 2020). We further create new synthetic datasets to verify the generalizability of the claims made by the authors.

3.3.1 Original Paper Datasets

NBA: Nodes represent basketball players. Features include nationality, age, salary, and several performance statistics. The edges are based on whether players follow each other on social media. The sensitive feature is nationality, and the classification task is to predict whether a player's salary is above the median.

Pokec: The data comes from a popular social network in Slovakia, and is separated into two subsets by region: Pokec-z and Pokec-n. The nodes in the graph are the users, and the features are information about the users. The region of the users is treated as the sensitive feature. The classification task is to predict the working field of the users.

3.3.2 Additional Datasets

To test the claim that Graphair works well regardless of graph structure and connectivity, we create a synthetic dataset (in five variants, to be explained below) using the DPAH module (Espín-Noboa et al., 2022), composed of people as nodes and their friendships as edges. The dataset has a majority group and a minority group, the latter of which might, for example, correspond to a racial minority. Each node has three attributes:

- Group membership (binary, *majority* or *minority*) – 70% of the nodes are majority while 30% are minority.
- Education (binary, *good* or *bad*) – 70% of majority nodes have good education, while only 30% of minority nodes have good education.
- Income (binary, *high* or *low*) – 90% of majority nodes have high income, while only 10% of minority nodes have high income.

Hence, education correlates with group membership, while group membership in turn correlates with income.

The task for a GNN classifier is to predict the income based on group membership and education. Graphair aims at not only deleting the sensitive attribute (which would be trivial) but also completely decorrelating it

from other attributes, as it is possible to recover sensitive attributes from other attributes (Sweeney, 2002). In this case, the education attribute can be used to predict the group membership with high accuracy, as the two correlate. From there on, the classifier would be able to deduce income. Therefore, it would be detrimental if Graphair only deleted the attribute "group membership", as the classifier could still easily learn to represent group membership as a latent variable and deduce income from it. In other words, if Graphair were able to fully decorrelate the sensitive feature from other features, it would need to fully mask both the education and the group membership attribute. That way, the classifier would only get nonsense information and have no other optimal choice than always output "high income" no matter the input, for which the accuracy would go to 66% (the percentage of high-income nodes in the dataset, i.e., the prior probability of having high income). Simultaneously, the classifier would achieve perfect fairness in this case. While in normal scenarios, masking the entire dataset input would not be useful, we create this isolated setting in which we can clearly assess the claims of Graphair as all dataset features correlate with the sensitive attribute.

Having created this isolated test setting, we now vary graph connectivity. We want to test the generalizability of Graphair to differently connected graph datasets, as in real-world settings, e.g., minority groups may be segregated in different ways. Specifically, if nodes connect often within their group but scarcely between groups, or vice-versa, Graphair should still work fine. This tendency to connect within or amongst groups is known as homophily (Espín-Noboa et al., 2022). The way nodes connect determines the information flow within the GNN and thus influences the prediction of a GNN classifier. We test whether (1) Graphair is capable of decorrelating the features at all in this setting and (2) whether this decorrelation breaks for certain graph connectivities (homophilies). We use a range of different graph connectivity configurations, creating five variations of the synthetic dataset:

1. Low connectivity both inside minority and majority group (called Syn-0.2-0.2). In this case, most friendships are between minority and majority people.
2. Low connectivity inside minority group and high connectivity inside majority group (called Syn-0.2-0.8).
3. Medium connectivity both inside minority and majority nodes (called Syn-0.5-0.5). This implies medium connectivity between minority and majority nodes as well.
4. High connectivity inside minority group and low connectivity inside majority group (called Syn-0.8-0.2).
5. High connectivity both inside minority and majority group (called Syn-0.8-0.8). In this case, few friendships exist between minority and majority people.

E.g., for Syn-0.2-0.8, the first value (0.2) refers to the tendency of minority nodes to connect to each other (i.e., homophily), while the second one (0.8) denotes the tendency of majority nodes to connect to each other. Simultaneously, minority nodes connect to majority ones often in this case, while majority nodes seldom connect to minority nodes. Thus, in this particular example, there are few friendships between minority and minority persons, and many between majority and majority persons. For further information on homophily, we refer to appendix H.

Table 1 shows all relevant dataset statistics, as well as the specific homophily settings used for the synthetic dataset creation.

3.4 Experimental setup

3.4.1 Original Experiment

Claim 1: Following the methodology of the authors, we conduct a grid search over the values 0.1, 1, and 10 to find the hyperparameter setting for α, γ, and λ that results in the best model in terms of average accuracy over 5 evaluation runs. The experiment is repeated for all three datasets: NBA, Pokec-z, and Pokec-n. Due to the size of Pokec datasets being significantly larger than others, batch training is used, where, following

Table 1: Dataset statistics. Synthetic datasets are made up of 700 majority and 300 minority nodes. The first number in the synthetic dataset names refers to the homophily of minority nodes (also shown in the second last column) while the second number refers to the homophily of majority nodes (also shown in the last column).

Dataset	Nodes	Node features	Edges	Inter-group edges	Intra-group edges	min. Hom.	maj. Hom.
NBA	403	39	16,570	4,401	12,169		
Pokec-z	67,797	59	882,765	39,804	842,961	-	-
Pokec-n	66,569	59	729,129	31,515	697,614		
Syn-0.2-0.2				23,695	6,275	0.2	0.2
Syn-0.2-0.8				20,470	9,500	0.2	0.8
Syn-0.5-0.5	1,000	3	29,970	12,557	17,413	0.5	0.5
Syn-0.8-0.2				9,107	20,863	0.8	0.2
Syn-0.8-0.8				5,841	24,129	0.8	0.8

the method described in (Zeng et al., 2020), one batch is created using random walk sampling with 1000 root nodes and walk length 3. This procedure results in batches of at most 3000 nodes.

Claim 2: The verification of this claim is addressed in an ablation study: we run our evaluation protocol while disabling the respective components of the augmentation module individually. Appendix 6 shows a graphic of this alteration.

Claim 3: To address this claim, we investigate how node-wise sensitive homophily (i.e., the tendency of nodes to connect with nodes having the same sensitive attribute value) and Spearman correlation between the sensitive and other features decrease in the fair setting.

3.4.2 Change of Experimental Setup for Better Robustness Assessment

As described in Section 3.2, to evaluate the Graphair framework the authors train the models f, k, and g once in an unsupervised manner, and then train a classifier 5 times to obtain the mean and variance of the results. However, this procedure does not show how robust and stable the actual proposed method (Graphair) is, but only how much variation in results the simple classifier undergoes throughout different runs. A more appropriate approach would be to test the stability of Graphair itself. Therefore, we conduct a second experiment, where models f, k, and g are trained five times, together with the classifier. All further experiments use this approach. Diagrams of the two evaluation protocols are shown in appendix A.

3.4.3 Longer Training Procedure

In the original paper, the authors claim that Graphair is trained on Pokec datasets for 500 epochs. However, upon reviewing the codebase, we discovered that the code did not implement mini-batch training correctly, resulting in the term "epochs" being used inaccurately to refer to steps.[3] We contacted the authors to obtain the appropriate parameters for running the code, and they supplied parameters corresponding to 500 steps of training.

Given that the full dataset contains approximately 66,000 nodes and mini-batches of up to 3000 nodes are used, it would require 22 steps to cover the entire dataset once. Therefore, to train the model for 500 epochs, the total number of steps should be equal to 11,000 ($= 500 \cdot 22$).

To thoroughly evaluate Graphair, we train the model for both 500 and 11,000 steps. Due to computational limitations, we conduct a grid search as detailed in Section 3.4.1 using 500 steps and subsequently re-run

[3]We contacted the authors about this bug. Since then, it has been fixed by the authors.

the experiments with the optimal hyperparameters for 11,000 steps. Unless specified otherwise, all further experiments use the models obtained from this longer training procedure.

3.4.4 Further Experiments to Verify Claims

To further test whether the proposed method improves fairness in comparison to unfair methods (**Claim 1**), we carry out two additional experiments.

Examining the effects of adversarial training: To validate the effect of adversarial training on the performance of the model, we train Graphair with hyperparameter $\alpha = 0$, which corresponds to not using the adversarial loss. Under such a setting, we expect fairness to deteriorate, and accuracy to improve (compared to training with a nonzero value like $\alpha = 1$) because the augmentation module no longer makes use of adversarial training to satisfy the fairness property. Appendix 7 shows a visual overview of this alteration.

Comparison to unfair baseline: Graphair consists of two parts: a fairness step which produces a fair graph, and a finetuning step which trains a classifier on that graph. The novelty of Graphair lies in the first step, while the second step is a standard downstream performance evaluation procedure. Even though the authors claim that their method results in much fairer graphs compared to the other graph fairness frameworks, they never compare their method against the "unfair" baseline.

To assess the actual benefits of employing Graphair, compared to not doing any fairness training, we turn off the fairness step and train the classifier on the original graph directly. This serves as a sanity check for the method, as we expect Graphair's fairness step to lead to improved fairness as compared to the unfair baseline. Appendix 8 shows a visual overview of this alteration.

3.4.5 Additional Experiments to Verify Graphair's Generalizability

To test the generalizability of the proposed method, we conduct the following experiments:

Generalizability to different GNN architectures: To investigate whether the method generalizes beyond GCNs to other Message-Passing GNNs (Gilmer et al., 2017), we modify the experimental setup, replacing all GCNs with Graph Attention Networks (GATs; Veličković et al., 2017). This verifies **Claim 1** in regards to whether the high accuracy and good fairness results hold in this case. Similarly to 3.4.1 we perform a full grid search to find the best hyperparameters.

Generalizability to graphs with varying homophily: We examine whether the method generalizes to different graph structures. To that end, we repeat the experiment on all five synthetic datasets described in Section 3.3.

3.5 Computational Requirements

We used a single NVIDIA RTX A6000 GPU for running the experiments. On this GPU, the average Graphair runtime per hyperparameter setting is around 3 minutes for models trained on Pokec datasets for 500 steps, 30 minutes for models trained for 11,000 steps, and around 30 seconds for models trained on the NBA dataset. In total, we trained 650 models. All experiments combined took a total of 119 GPU hours.

4 Results

4.1 Results Verifying Original Claims

Claim 1 (group fairness improvement and similar accuracy): In table 2, we show the results of running a grid search with the original paper's evaluation protocol and our proposed one, as described in section 3.2. In addition, we report the results of a longer training procedure on both Pokec datasets. On the NBA dataset, the results are fully reproduced with both evaluation protocols. In some cases, measurements even show a slight improvement in accuracy and fairness metrics. The accuracy on the Pokec datasets drops by several percentage points (around 5% for Pokec-z and 4% for Pokec-n), while the fairness is close

Table 2: The results on the NBA, Pokec-z, and Pokec-n datasets, showing the original and the reproduced results with an identical experimental setup, the results employing our proposed evaluation protocol, and the longer training results.

DATASET	SETTING	EVALUATION PROTOCOL	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	original	original	69.36 ± 0.45	2.56 ± 0.41	4.64 ± 0.17
	reproduction	original	$\mathbf{69.64 \pm 0.69}$	$\mathbf{0.14 \pm 0.00}$	3.98 ± 0.87
	reproduction	ours	69.36 ± 0.46	2.57 ± 1.13	$\mathbf{2.33 \pm 1.53}$
Pokec-z	original	original	$\mathbf{68.17 \pm 0.08}$	2.10 ± 0.17	2.76 ± 0.19
	reproduction	original	61.83 ± 0.29	$\mathbf{1.48 \pm 0.79}$	3.97 ± 1.06
	reproduction	ours	60.60 ± 1.97	2.84 ± 2.04	3.59 ± 1.96
	reproduction (longer training)	ours	63.39 ± 0.68	2.28 ± 1.52	$\mathbf{2.24 \pm 1.99}$
Pokec-n	original	original	$\mathbf{67.43 \pm 0.25}$	$\mathbf{2.02 \pm 0.40}$	$\mathbf{1.62 \pm 0.47}$
	reproduction	original	63.58 ± 0.30	3.80 ± 0.43	3.83 ± 0.60
	reproduction	ours	62.64 ± 0.84	3.82 ± 2.99	3.98 ± 2.60
	reproduction (longer training)	ours	63.88 ± 0.91	2.85 ± 0.59	2.40 ± 1.14

Table 3: Setting alpha to zero, compared to its optimal value obtained through the grid search.

DATASET	α	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	optimal	$\mathbf{69.36 \pm 0.45}$	$\mathbf{2.57 \pm 1.13}$	$\mathbf{2.33 \pm 1.53}$
	0	69.08 ± 1.71	5.09 ± 2.30	5.26 ± 2.54
Pokec-z	optimal	$\mathbf{63.39 \pm 0.68}$	2.28 ± 1.52	2.24 ± 1.99
	0	62.87 ± 1.22	$\mathbf{1.70 \pm 0.6}$	$\mathbf{2.14 \pm 1.64}$
Pokec-n	optimal	63.88 ± 0.91	$\mathbf{2.85 \pm 0.59}$	$\mathbf{2.40 \pm 1.14}$
	0	$\mathbf{64.57 \pm 0.48}$	4.72 ± 1.44	4.34 ± 1.91

to the original results. It is important to note that the variance in the runs with our evaluation protocol is substantially higher in nearly all cases, especially on the fairness metrics. This indicates instability in Graphair's results. We present the results achieved using the hyperparameters that lead to optimal fairness. We report primarily these results because, under the settings optimized for the highest accuracy, the fairness was notably worse. The best accuracy results can be found in appendix B and accuracy-fairness trade-off plots can be found in appendix C.

Table 3 shows the results of disabling adversarial training as described in section 3.4.4. For NBA and Pokec-n datasets, we observe higher DP and EO values (at least twice as high compared to the full method), which is expected. However, the results on the Pokec-z dataset show that the network without adversarial training leads to similar or even lower values compared to the full model. This indicates that the adversarial training does not lead to better graph fairness on this dataset, which again shows Graphair's instability. Therefore, this experiment provides only partial evidence for the reproducibility of **Claim 1**.

Table 4: Comparison of Graphair to not using fairness training.

DATASET	SETTING	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	full Graphair method	$\mathbf{69.36 \pm 0.45}$	$\mathbf{2.57 \pm 1.13}$	$\mathbf{2.33 \pm 1.53}$
	w/o fairness training	68.65 ± 2.03	4.82 ± 1.93	2.93 ± 2.41
Pokec-z	full Graphair method	63.39 ± 0.68	$\mathbf{2.28 \pm 1.52}$	$\mathbf{2.24 \pm 1.99}$
	w/o fairness training	$\mathbf{70.21 \pm 0.49}$	8.93 ± 1.41	8.53 ± 1.45
Pokec-n	full Graphair method	63.88 ± 0.91	2.85 ± 0.59	2.40 ± 1.14
	w/o fairness training	$\mathbf{68.51 \pm 0.29}$	$\mathbf{1.37 \pm 0.79}$	$\mathbf{1.27 \pm 0.95}$

Table 5: Comparisons among different components in the augmentation model. FM stands for Feature Masking, and EP stands for Edge Perturbation.

DATASET	SETTING	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	Graphair w/o FM	68.79 ± 1.19	3.33 ± 2.43	2.82 ± 2.31
	Graphair w/o EP	$\mathbf{71.06 \pm 3.09}$	6.32 ± 1.46	3.65 ± 1.73
	Graphair	69.36 ± 0.45	$\mathbf{2.57 \pm 1.33}$	$\mathbf{2.33 \pm 1.53}$
Pokec-z	Graphair w/o FM	63.03 ± 0.74	$\mathbf{0.85 \pm 0.85}$	$\mathbf{0.79 \pm 0.96}$
	Graphair w/o EP	$\mathbf{68.15 \pm 0.60}$	9.18 ± 3.06	9.08 ± 3.13
	Graphair	63.39 ± 0.68	2.28 ± 1.52	2.24 ± 1.99
Pokec-n	Graphair w/o FM	64.58 ± 0.86	4.35 ± 2.27	4.22 ± 1.71
	Graphair w/o EP	$\mathbf{66.22 \pm 1.29}$	2.59 ± 1.56	$\mathbf{2.18 \pm 1.49}$
	Graphair	63.39 ± 0.68	$\mathbf{2.28 \pm 1.52}$	2.24 ± 1.99

Table 4 shows the results of comparing Graphair to an unfair baseline, which we established by excluding the fairness component from the pipeline. Compared to Graphair, fairness is worse in the baseline setting on the NBA and Pokec-z datasets, but better on Pokec-n, providing again only partial evidence for **Claim 1**. Further discussion on these results can be found in section 5.

Claim 2 (augmenting both edges and node features improves performance): Table 5 shows the results of the ablation study. For the NBA dataset, **Claim 2** is reproducible. However, for Pokec-z, Graphair without node feature masking achieves surprisingly good fairness metrics, approximately three times better than those of the default Graphair. Lastly, on Pokec-n, Graphair without edge perturbation achieves better EO than default Graphair. Generally, disabling edge perturbation leads to the highest accuracy. Possible reasons for this behavior are discussed in section 5.

Claim 3 (Graphair reduces node-wise sensitive homophily and feature-sensitive correlation): Figures 1 and 2 show a recreation of figures 3 and 4 of the original paper. Figure 1 shows the same trend of decreasing node-wise sensitive homophily as in the original work. However, figure 2 exhibits far less decorrelation between the sensitive and the other features for the fair view of the original graph.

Table 6: Results for GAT network juxtaposed to the normal GCN setting.

DATASET	SETTING	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	GAT	63.97 ± 1.76	4.55 ± 1.92	13.05 ± 7.64
	GCN (normal)	$\mathbf{69.36 \pm 0.45}$	$\mathbf{2.57 \pm 1.13}$	$\mathbf{2.33 \pm 1.53}$
Pokec-z	GAT	60.94 ± 1.92	3.70 ± 2.87	4.86 ± 2.79
	GCN (normal)	$\mathbf{63.39 \pm 0.68}$	$\mathbf{2.28 \pm 1.52}$	$\mathbf{2.24 \pm 1.99}$
Pokec-n	GAT	59.72 ± 2.06	$\mathbf{1.78 \pm 1.21}$	$\mathbf{0.81 \pm 0.42}$
	GCN (normal)	$\mathbf{63.88 \pm 0.91}$	2.85 ± 0.59	2.40 ± 1.14

Table 7: Results for different artificial datasets.

DATASET	ACCURACY ↑	Δ DP ↓	Δ EO ↓
Syn-0.2-0.2	70.60 ± 1.74	23.47 ± 10.72	19.09 ± 10.52
Syn-0.2-0.8 Syn-0.5-0.5 Syn-0.8-0.2	66.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Syn-0.8-0.8	77.20 ± 5.88	50.93 ± 29.14	54.55 ± 31.49

4.2 Results Testing the Generalizability of the Graphair Method

Graph-Attention: Table 6 shows the results of replacing all GCNs with GATs. Performance is consistently slightly worse for the GAT variants. Specifically, the method fails to achieve the same accuracy and fairness on the NBA dataset, whereas the performance on all metrics is only slightly worse for the Pokec-z dataset. Interestingly, the model with GAT trained on the Pokec-n dataset achieves much better fairness metrics than the model trained with GCN, resulting in approximately 1.5 times improvement in DP and 3 times improvement in EO.

Synthetic Datasets: The results for the five different synthetic datasets can be seen in table 7. It shows high accuracy and low fairness for the dataset variants with homophily values of 0.2 and 0.2, as well as 0.8 and 0.8, respectively. The first setting implies that within-group connections are scarce while between-group connections are ample, whereas the latter setting implies ample within-group and low between-group connections. On the other hand, the other homophily settings achieve an accuracy equal to the prior of 0.66, as 66% of nodes exhibit high income. Simultaneously, fairness is optimal in these cases at 0.00 ΔDP and ΔEO for Syn-0.2-0.8, Syn-0.5-0.5, and Syn-0.8-0.2. The homophily in these settings could be defined as more harmonized, and the method achieves full decorrelation of the output and the sensitive feature by also masking the third feature, which in turn is correlated with the sensitive feature.

5 Discussion

Reproducibility: Although the original paper does not test the entire method in a multi-run setting with different random seeds, **Claim 1** and **Claim 2** are largely reproduced using both the original and our proposed evaluation setup. However, both claims could not be fully reproduced for all datasets. In our experiments, Graphair does not produce better results on the Pokec-n dataset as compared to baselines

that do not account for fairness. The ablation study is also not fully reproduced, as for the Pokec-z dataset, Graphair without node feature masking showed significantly better fairness than default Graphair. Moreover, for the Pokec-n dataset, Graphair without edge perturbation produced better results than the full method. In addition to that, **Claim 3** could only be partially reproduced, as node-wise sensitive homophily was shown to decrease, but Spearman correlation between the sensitive feature and other features did not significantly decrease in our experiments. We conclude that **Claim 1**, **Claim 2** and **Claim 3** are partially reproducible. We note that throughout multiple experiments, we observe that the method is not stable, which could hurt its adoption for a variety of tasks.

Generalizability: We conduct several additional experiments to verify the generalizability of the method. Firstly, we show that other message-passing GNN architectures perform reasonably well on the two larger datasets (Pokec-z and Pokec-n) while the results are considerably worse for the NBA dataset. These results are in range, as attention networks are known to have a necessity for significantly more training data than convolutional networks (Dosovitskiy et al., 2020). We conclude that the method generalizes to other message-passing architectures and thus exhibits robustness regarding the graph architecture used.

Secondly, we show that the method does not generalize to graphs with varying homophily. Our five synthetic datasets exhibit different levels of connectivity within groups and in between groups. On one hand, three of our datasets have harmonized connectivity, i.e., if intra-group connections are scarce, inter-group connections are plenty, or both are balanced. On the other hand, there are two extreme variations where there are extremely scarce connections within and in between groups, or many intra- and inter-connections. Specifically, the method can achieve full decorrelation for harmonized homophily values for the majority and minority groups, while extreme values (very low inter-group or intra-group connectivity) lead to failure. This shows that not every dataset could be used in conjunction with Graphair to achieve fairness, as harmonized homophily characteristics can leave substantial bias in the dataset even after augmentation through Graphair.

Instability: Using our updated evaluation procedure, the variance in all the results was significantly higher compared to the results reported in the original paper. Additionally, ablations reveal that fairness can actually be better on the Pokec-z datasets when ablating node feature masking or edge perturbation. Surprisingly, ablating node feature masking on Pokec-z leads to significantly increased fairness. We note that across all the performed experiments, Graphair exhibits inconsistent behavior, often leading to unexpected results on at least one dataset. Therefore, we conclude that Graphair is generally unstable, limiting its reproducibility and generalizability.

5.1 What was easy

The study design was mostly straightforward, with very clear and well-written explanations in the paper. The availability and ease of use of the code also contributed to this. That way, we were able to run the experiments described in the original paper without major problems. In addition, the claims were clearly presented in the original paper, aiding our work.

5.2 What was difficult

The code had several shortcomings that posed problems to a full reproduction. Firstly, random seeds were only set for the dataset shuffling procedures and evaluation, whereas the training of the networks in the Graphair method did not have the random seed set. Secondly, there were various hardcoded hyperparameter settings in the code, many of which were not mentioned in the paper. Lastly, some parts of the experimental code were challenging to execute, such as the mini-batching procedure which required extensive work to set up and run. Also, several bugs were identified in the code, some of which were fixed by the authors.

5.3 Communication with Original Authors

We sent multiple emails to the authors with clarification questions, to which the authors swiftly and kindly answered each time and with clear explanations. The authors moreover improved their code base after our emails, supporting our work.

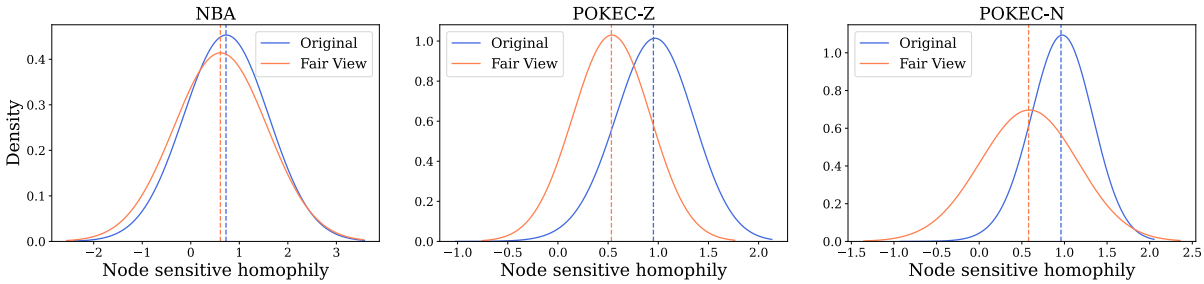

Figure 1: Distributions of node-wise sensitive homophily in the original and the fair graph data.

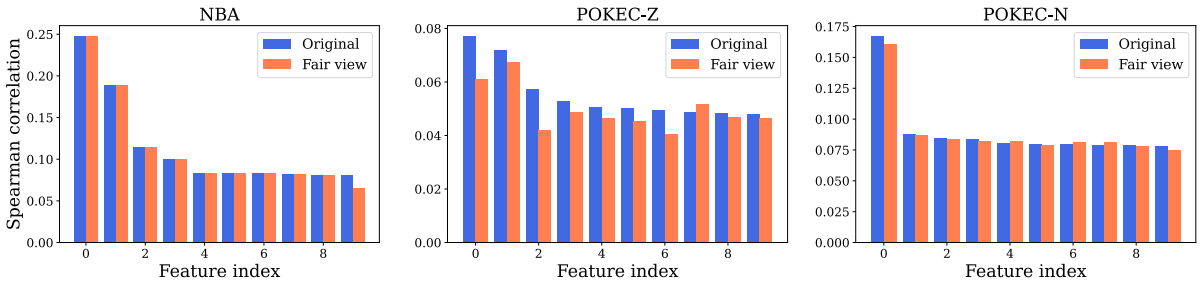

Figure 2: Spearman correlation between the sensitive feature and the other node features in the original and the fair graph data.

References

Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H Chi. Data decisions and theoretical implications when adversarially learning fair representations. *arXiv preprint arXiv:1707.00075*, 2017.

Enyan Dai and Suhang Wang. Fairgnn: Eliminating the discrimination in graph neural networks with limited sensitive attribute information. *CoRR*, abs/2009.01454, 2020. URL `https://arxiv.org/abs/2009.01454`.

Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 680–688, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Lisette Espín-Noboa, Claudia Wagner, Markus Strohmaier, and Fariba Karimi. Inequality and inequity in network-based ranking and recommendation algorithms. *Scientific reports*, 12(1):2012, 2022.

Hongyang Gao and Shuiwang Ji. Graph u-nets. In *international conference on machine learning*, pp. 2083–2092. PMLR, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pp. 1802–1808. AAAI Press, 2017. ISBN 9780999241103.

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

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pp. 8230–8248. PMLR, 2022.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2016.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in ml-based science. *arXiv preprint arXiv:2207.07048*, 2022.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

O Deniz Kose and Yanning Shen. Fair node representation learning via adaptive data augmentation. *arXiv preprint arXiv:2201.08549*, 2022.

Hongyi Ling, Zhimeng Jiang, Youzhi Luo, Shuiwang Ji, and Na Zou. Learning fair graph representations via automated data augmentations. In *The Eleventh International Conference on Learning Representations*, 2023.

Meng Liu, Cong Fu, Xuan Zhang, Limei Wang, Yaochen Xie, Hao Yuan, Youzhi Luo, Zhao Xu, Shenglong Xu, and Shuiwang Ji. Fast quantum property prediction via deeper 2d and 3d graph networks. *arXiv preprint arXiv:2106.08551*, 2021a.

Meng Liu, Youzhi Luo, Limei Wang, Yaochen Xie, Hao Yuan, Shurui Gui, Haiyang Yu, Zhao Xu, Jingtun Zhang, Yi Liu, et al. Dig: A turnkey library for diving into graph deep learning research. *Journal of Machine Learning Research*, 22(240):1–9, 2021b.

Yi Liu, Limei Wang, Meng Liu, Yuchao Lin, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2022.

Christos Louizos, Kevin Swersky, Yujia Li, Max Welling, and Richard Zemel. The variational fair autoencoder. *arXiv preprint arXiv:1511.00830*, 2015.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 27(1):415–444, 2001.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, 54(6):1–35, 2021.

Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kıcıman. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in big data*, 2:13, 2019.

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

Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International conference on machine learning*, pp. 12241–12252. PMLR, 2021.

Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=BJe8pkHFwS`.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.

A Evaluation Protocol

.

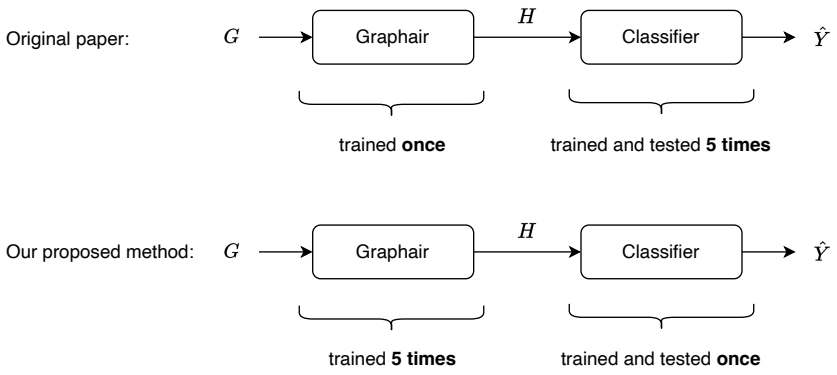

Figure 3: Evaluation protocol of the original paper and our proposed method.

B Grid Search Best Accuracy Results

Table 8: Best-accuracy results on the NBA, Pokec-z, and Pokec-n datasets, showing the original ones, the reproduced results with an identical experimental setup, and the results employing our proposed evaluation protocol.

DATASET	SETTING	ACCURACY ↑	Δ DP ↓	Δ EO ↓
NBA	original (original evaluation protocol)	69.36 ± 0.45	$\mathbf{2.56 \pm 0.41}$	4.64 ± 0.17
	reproduction (original evaluation protocol)	$\mathbf{70.92 \pm 1.19}$	18.50 ± 5.32	9.30 ± 3.63
	reproduction (our evaluation protocol)	70.35 ± 0.53	4.75 ± 2.60	$\mathbf{4.55 \pm 1.95}$
Pokec-z	original (original evaluation protocol)	$\mathbf{68.17 \pm 0.08}$	$\mathbf{2.10 \pm 0.17}$	$\mathbf{2.76 \pm 0.19}$
	reproduction (original evaluation protocol)	62.01 ± 0.03	3.74 ± 0.69	4.93 ± 1.01
	reproduction (our evaluation protocol)	60.94 ± 2.34	3.73 ± 1.43	4.36 ± 2.68
Pokec-n	original (original evaluation protocol)	$\mathbf{67.43 \pm 0.25}$	$\mathbf{2.02 \pm 0.40}$	$\mathbf{1.62 \pm 0.47}$
	reproduction (original evaluation protocol)	63.59 ± 0.30	3.80 ± 0.43	3.83 ± 0.60
	reproduction (our evaluation protocol)	62.79 ± 1.41	5.48 ± 2.62	4.73 ± 1.87

C Pareto-Fronts of Grid Searches

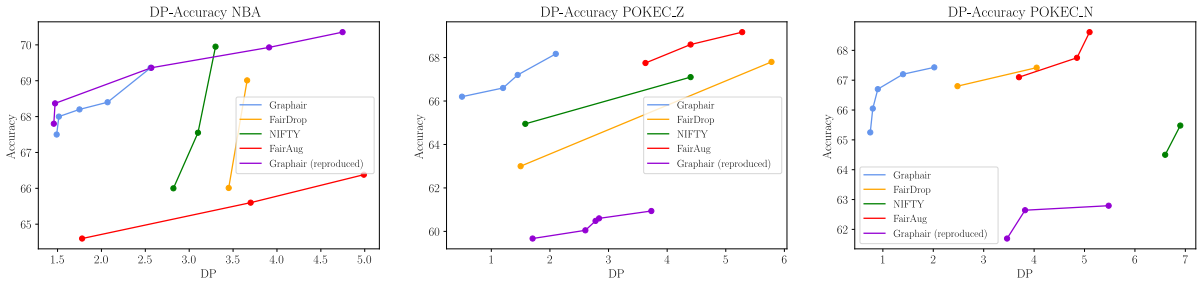

Figure 4: DP-Accuracy Pareto front plots for all three datasets. Graphair (reproduced) refers to the model trained with our proposed evaluation protocol.

D Original Framework

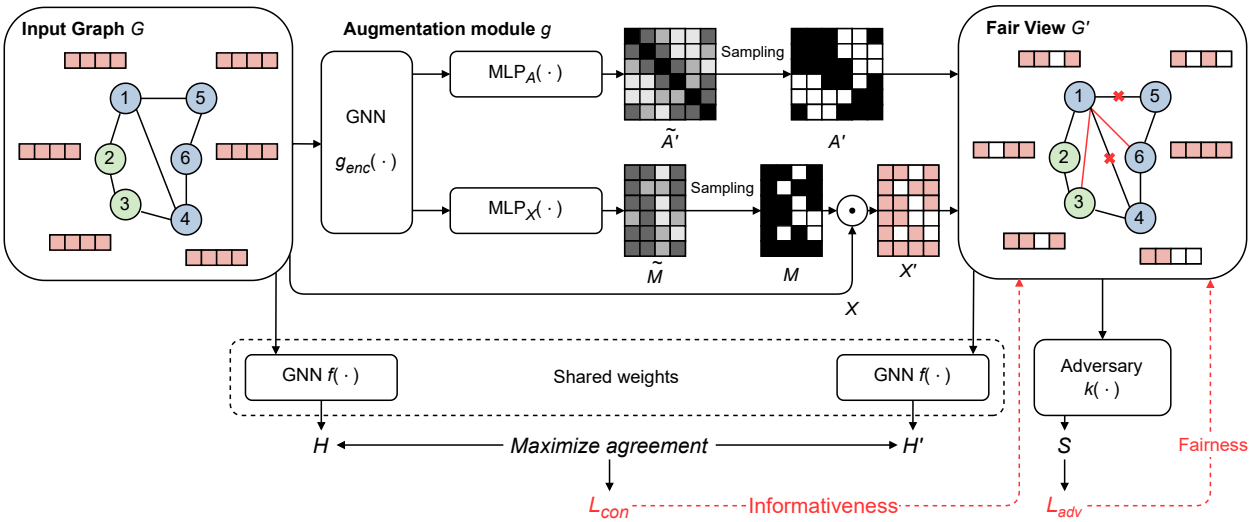

Figure 5: Graphair framework. The graphic was adapted from the original paper (Ling et al., 2023).

E Framework for Ablation Study

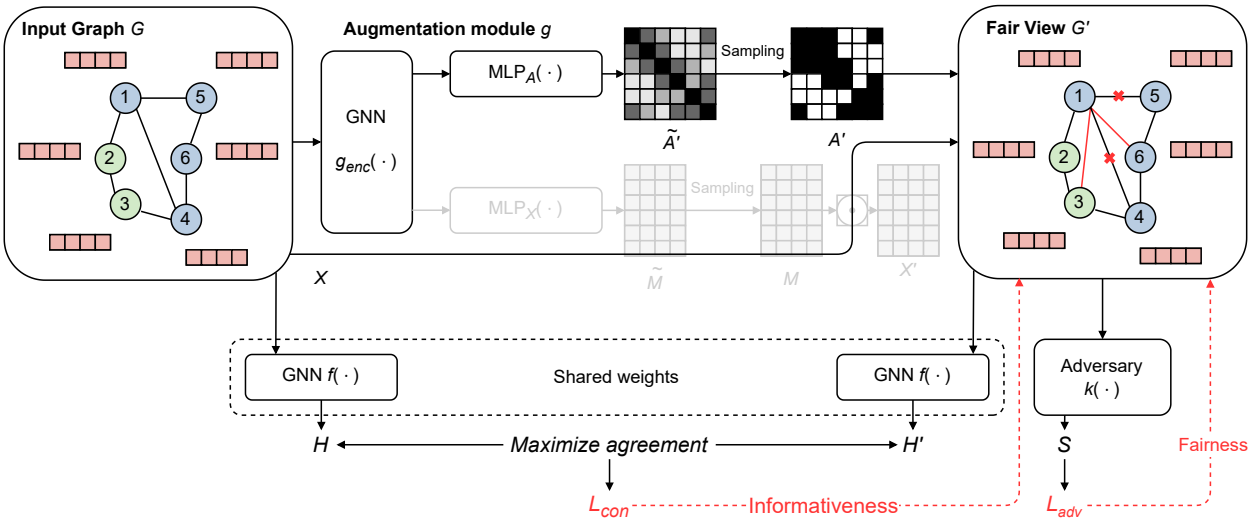

Figure 6: Graphair framework without node feature masking for the ablation study.

F Framework without Adversarial Training

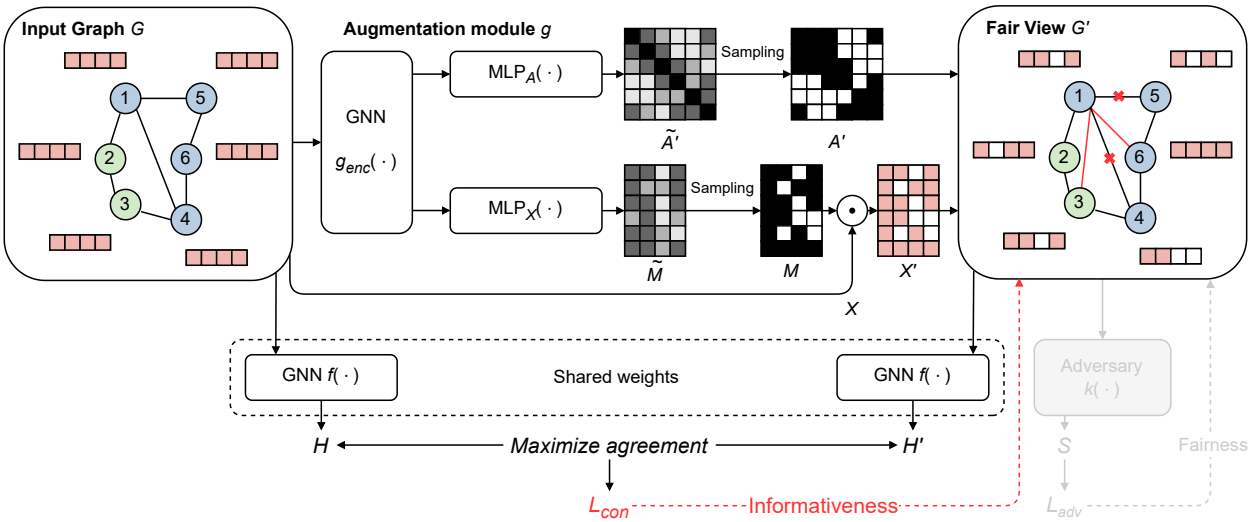

Figure 7: Graphair framework without the adversarial training.

G Framework for Supervised Training

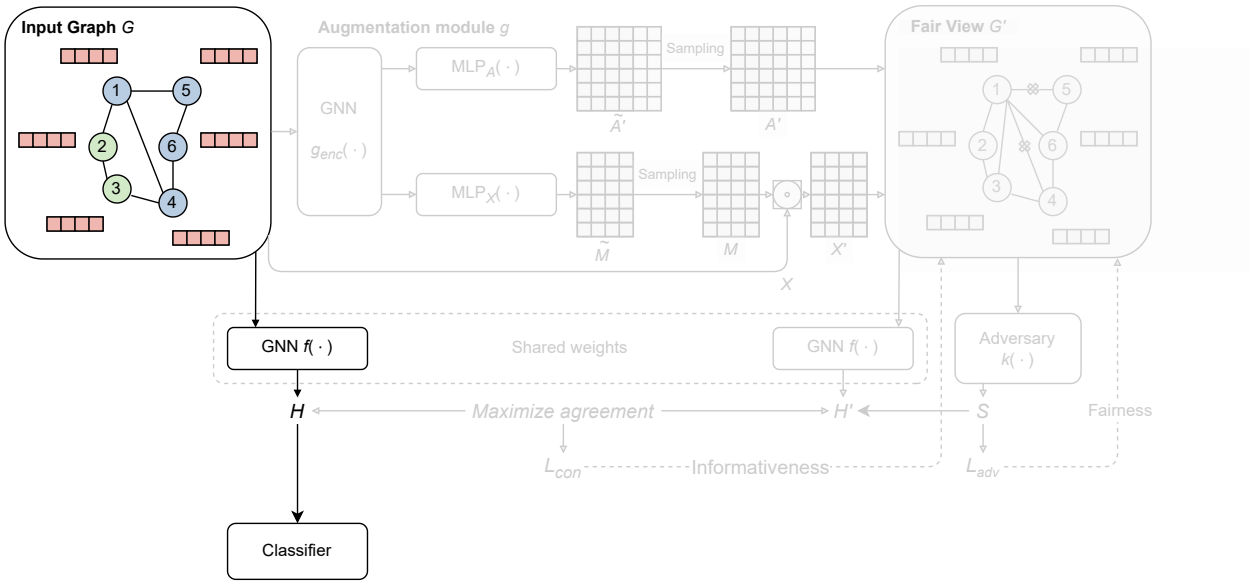

Figure 8: Fully-supervised training without Graphair framework.

H Further Explanations on Homophily

Homophily refers to the tendency of similar nodes to connect (McPherson et al., 2001). Hence, the probability that two nodes i and j connect is determined by the homophily value between their classes c_i and c_j, where a value lower than 0.5 means that i is more likely to connect to a node from a different class than c_j, and a value higher than 0.5 means that i is more likely to connect to a node of class c_j than other classes. The specific definition of the corresponding probabilities is defined in the work by Espín-Noboa et al. (2022). Since we use two groups, i.e., majority M and minority m, there are two homophily settings: h_{MM} referring to the tendency of majority group nodes connecting to each other, and h_{mm} referring to how much minority groups tend to connect to each other. Note that the between-group connectivity values are determined correspondingly: $h_{mM} = 1 - h_{mm}$ and $h_{Mm} = 1 - h_{MM}$.

I Additional Results for Scaled Training Runs

In addition to repeating the experiments on the Pokec datasets with corrected step counts (11,000 instead of 500), we perform experiments with larger batch sizes. These are performed using the initial step count in light of computational resource limits. Moreover, due to the maximum memory capacity of the utilized GPU of 40GB, the largest possible batch size we could fit was 2000.
The results are shown in Table 9 alongside the ones for larger step counts. Larger batch size generally performs worse on accuracy, DP, and EO. However, on Pokec-z, a lower DP is achieved compared to the other large-scale run.

Table 9: Results for our scaled training runs on the Pokec datasets with significantly larger epoch count and batch size. The NBA dataset is omitted as its small size makes an analogous run superfluous.

DATASET	**BATCH SIZE**	**EPOCHS**	**ACCURACY**	**Δ DP $\downarrow$**	**Δ EO $\downarrow$**
Pokec-z	1000	11000	63.39 ± 0.68	2.28 ± 1.52	2.24 ± 1.99
Pokec-z	2000	500	60.46 ± 2.58	1.84 ± 0.26	2.56 ± 1.00
Pokec-n	1000	11000	63.88 ± 0.91	2.85 ± 0.59	2.40 ± 0.11
Pokec-n	2000	500	62.59 ± 1.30	5.85 ± 0.65	5.31 ± 1.23