Reproducibility Study Of Learning Fair Graph Representations Via Automated Data Augmentations

Abstract

In this study, we undertake a reproducibility analysis of "Learning Fair Graph Representations Via Automated Data Augmentations" by Ling et al. (2022). We assess the validity of the original claims centered around node classification tasks and explore the performance of the Graphair framework in link prediction tasks. Our investigation reveals that while we can partially reproduce some of the original claims—likely impeded by unstable training and a code bug identified through collaboration with the original authors—we fully substantiate another claim. Additionally, we broaden the application of Graphair from node classification to link prediction across various datasets. This expansion demonstrates Graphair's superior performance in fairness metrics when compared to existing models, showing only a slight reduction in accuracy. This underlines Graphair's potential applicability in a wider array of graph-based learning contexts, showcasing its capability to maintain high fairness standards without significantly compromising accuracy. Our code base can be found on GitHub https://anonymous.4open.science/r/Reproducibility-Study-Of-Graphair-1DB6.

1 Introduction

Graph Neural Networks (GNNs) have become increasingly popular for their exceptional performance in various applications (Hamaguchi et al., 2017; Liu et al., 2022; Han et al., 2022a). A key application area is graph representation learning (Grover & Leskovec, 2016; Hamilton, 2020; Han et al., 2022b), where a significant concern is the potential for GNNs to inherit or amplify biases present in the graph representation training data. This can lead to discriminatory model behavior (Dai & Wang, 2022).

To address this issue, Ling et al. (2022) introduced Graphair, an automated data augmentation technique aimed at learning fair graph representations without maintaining biases from the training data.

This work evaluates the main claims made by Ling et al. (2022), which were based solely on the performance of the framework in node classification tasks. We expand our evaluation by conducting additional experiments to assess the adaptability and generalizability of Graphair through its application to a different downstream task, namely, link prediction. This approach allows us to further test the performance and fairness of the embeddings.

For this purpose, we apply Graphair to new real-world datasets, adapting certain aspects of the framework and fairness metrics to suit link prediction. Our contributions are summarized as follows:

- Replicating the original experiments to assess the reproducibility of the primary claims. We find that we can only partially verify two of the three claims made by the authors which is likely due to unstable training and a bug in the code acknowledged by the authors. The remaining claim is fully verified.

- Applying Graphair to various real-world datasets for link prediction, which necessitated adjustments to the framework and fairness metrics. This endeavor provides insights into Graphair's adaptability and generalizability to another downstream task.

2 Scope of reproducibility

The original paper *Learning Fair Graph Representations Via Automated Data Augmentations* by Ling et al. (2022) introduces Graphair, an innovative automated graph augmentation method for fair graph representation learning. This approach stands out from prior methods (Luo et al., 2022; Zhao et al., 2022; Agarwal et al., 2021) by utilizing a dynamic and automated model to generate a new, fairer version of the original graph, aiming to balance fairness and informativeness (Ling et al., 2022).

In this work, we study the reproducibility of this paper. Besides examining the three main claims, we also assess the adaptability and effectiveness of Graphair in a different context, namely link prediction. This extension tests the framework's ability to maintain fairness and informativeness in a different downstream tasks. We will test this on a variety of datasets. The claims and our extension are as follows:

- Claim 1: Graphair consistently outperforms state-of-the-art baselines in node classification tasks for real-world graph datasets. Our extension investigates if this superiority also holds in the context of link prediction tasks.
- Claim 2: Both fair node features and graph topology structures contribute to mitigating prediction bias.
- Claim 3: Graphair can automatically generate new graphs with fair node topology and features.

3 Methodology

3.1 Graphair

The *Graphair* framework introduces a novel approach to graph augmentation, focusing on the dual objectives of maintaining information richness and ensuring fairness. It incorporates a model g, which adeptly transforms an input graph $G = \{A, X, S\}$, where A represents the adjacency matrix, X the node features, and S the sensitive attributes, into an augmented counterpart $G' = \{A', X', S\}$. This transformation utilizes two main operations: T_A for adjusting the adjacency matrix A and T_X for masking node features X. A GNN-based encoder g_{enc} precedes these transformations, tasked with extracting deep embeddings to inform these operations. The resulting augmented graph G' is designed to capture the core structural and feature-based elements of the original graph G, while simultaneously keeping fairness principles to prevent the propagation of sensitive attribute information. This objective is achieved by training an adversarial model and using contrastive loss to optimize G'. An overview of the Graphair framework is presented in Figure 3 in Appendix A.1.

3.1.1 Adversarial Training for Fairness

To ensure the fairness of the augmented graph, the model employs an adversarial training strategy (Eq. 1). The adversarial model k learns to predict sensitive attributes S from the graph's features. Simultaneously, the representation encoder f and augmentation model g are optimized to minimize the predictive capability of the adversarial model, effectively removing biases from the augmented graph. The primary goal of adversarial training is to guide the encoder in generating representations that are free from sensitive attribute information. Here, $\hat{S}_i$ represents the predicted sensitive attributes, and n is the number of nodes.

$$\min_{g,f} \max_{k} L_{\text{adv}} = \min_{g,f} \max_{k} \frac{1}{n} \sum_{i=1}^{n} \left[S_i \log \hat{S}_i + (1 - S_i) \log(1 - \hat{S}_i) \right] \tag{1}$$

3.1.2 Contrastive Learning for Informativeness

The model employs contrastive learning (see Eq. 2) to ensure that the augmented graphs are fair and informative. By aligning the representations of nodes between the input and augmented graphs, the model strives to retain essential information content. The aim of the contrastive learning approach is to ensure a high degree of similarity between the node representations h_i of the original and h'_i of the augmented graphs. The function L_{con} denotes the contrastive loss between two node representations.

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

3.1.3 Integrated Training Objective

The training framework of *Graphair* integrates aforementioned adversarial and contrastive learning objectives in a coherent manner. This integrated approach ensures that the encoder not only minimizes bias in the graph representations but also captures the essential attributes of the original graph. Through this methodology, *Graphair* seeks to produce augmented graphs that are less susceptible to bias, contributing to fairer graph representation learning while retaining the valuable information contained in the data. The overall training process is described as the following min-max optimization procedure, where α, β, and γ are hyperparameters balancing the different loss components, including adversarial loss L_{adv}, contrastive loss L_{con}, and a reconstruction loss L_{reconst}:

$$\min_{f,g} \max_{k} L = \min_{f,g} \max_{k} \alpha L_{\text{adv}} + \beta L_{\text{con}} + \gamma L_{\text{reconst}} \tag{3}$$

3.2 Datasets

For reproducing the main claims, we employ the same datasets as the original paper by Ling et al. (2022), including dataset splits and sensitive and target attributes. We use three real-world graph datasets: NBA[1], which contains player statistics, and two subsets of the Pokec social network from Slovakia, specifically Pokec-n and Pokec-z (Dai & Wang, 2021). The details of these datasets are summarized in Table 1.

For the link prediction task, we utilize commonly used datasets in this domain: Cora, Citeseer, and Pubmed. In these datasets, nodes represent scientific publications, and edges indicate whether one publication references another. Notably, these datasets feature a more extensive range of features compared to those used by Ling et al. (2022). Further details are provided in Table 1, where the size of the set of possible sensitive attributes, denoted as $|S|$, is indicated. For all datasets, the sensitive attribute is the paper class. The dataset splits and sensitive attributes in our study are consistent with those utilized in (Spinelli et al., 2021), aiming to benchmark our findings against theirs. Spinelli et al. (2021) focuses on improving fairness in graph representation networks for link prediction tasks and is recognized, to our knowledge, for reporting state-of-the-art performance in fairness metrics, as detailed in Section 3.4, across these datasets.

Table 1: Dataset Statistics

Dataset	S	$\|S\|$	Features	Nodes	Edges
NBA	nationality	2	39	403	16,570
Pokec-z	region	2	59	67,797	882,765
Pokec-n	region	2	59	66,569	729,129
Citeseer	paper class	6	3,703	3,327	9,104
Cora	paper class	7	1,433	2,708	10,556
PubMed	paper class	3	500	19,717	88,648

3.3 Hyperparameters

Node Classification To align our experiments closely with the original study, we adopt the hyperparameters as specified by the authors. This includes a grid search on the hyperparameters α, γ, and λ, and we expand

[1] https://www.kaggle.com/datasets/noahgift/social-power-nba

this search to include β, which is not originally tuned. The grid search is conducted on these hyperparameters with the values $\{0.1, 1.0, 10.0\}$.

For training the adversarial model k, augmentation encoder g, representation encoder f, and the evaluation classifier, we apply the hyperparameters detailed in the original study. We use the default settings from the original code where specific hyperparameters are not disclosed. This decision is verified by the authors as correct. A comprehensive list of all hyperparameters is provided in Table 8 in Appendix A.2.

Link Prediction We replicate the grid search from the node classification for link prediction on the Citeseer, Cora, and PubMed datasets. We compare these results with baseline results from Spinelli et al. (2021). Based on preliminary results, we increase the epoch count to 200, the learning rate for the Graphair module to $1e-4$, and for the classifier, to $5e-3$. We keep the remaining hyperparameters unchanged.

3.4 Experimental Setup and Metrics

We default to the settings of the original authors to replicate their findings. We obtain the model's codebase from the authors via the DIG library, specifically using the FairGraph module[2]. Due to the large number of nodes for the PubMed and Pokec datasets, we use the GraphSAINT framework. This framework allows for the implementation of mini-batching, reducing the computational complexity when performing training and evaluation on Graphair. An implementation of the framework can bound as part of the PyG library[3]. To enhance reproducibility, we employ complete seeding across all operations, which was missing in some operations of the original code.

To verify Claims 1 and 2, we follow the procedure described by Ling et al. (2022), with the addition of our expanded hyperparameter search detailed in Section 3.3. For Claim 3, due to time and memory constraints, we compute the homophily and Spearman correlation values for the Pokec datasets on a mini-batch instead of the entire graphs. Aside from this, we adhere to the same procedures for this claim as well.

3.4.1 Link Prediction

In our study, we extend the scope of the original work to encompass link prediction. We implement several modifications to adjust the Graphair network for the downstream task of link prediction. Initially, we modifiy the output size of the adversarial network from two, which corresponded to the binary sensitive feature in the Pokec and NBA datasets, to the respective number of distinct values of the sensitive feature in the corresponding dataset. The classification network is not modified.

To create the link embeddings for the input of the classification network, we compute the Hadamard product of the node embeddings Horn & Johnson (2012), which pairs node embeddings effectively. For nodes v and u, we define the link embedding h_{vu} as:

$$h_{vu} = h_v \circ h_u \tag{4}$$

These new link embeddings require labels that reflect the sensitive attributes of the connected nodes. To this end, we integrate dyadic-level fairness criteria into our fairness assessment for link prediction. We form dyadic groups that relate sensitive node attributes to link attributes, following the mixed and subgroup dyadic-level fairness principles suggested by Masrour et al. (2020). The mixed dyadic-level groups classify links as either inter- or intra-group based on whether they connect nodes from the same or different sensitive groups. The subgroup dyadic-level approach assigns each link to a subgroup based on the sensitive groups of the nodes it connects. A subgroup is formed for each possible combination of sensitive groups. This method facilitates the measurement of fairness in two distinct aspects. Firstly, it assesses how well each protected subgroup is represented in link formation at the subgroup dyadic-level. Secondly, it evaluates node homogeneity for links at the mixed dyadic-level.

For assessing fairness in link prediction, we aim to optimize for Equality of Opportunity (EO) and Demographic Parity (DP) using these dyadic groups. Following the original paper's definition, DP is defined

[2] https://github.com/divelab/DIG/tree/dig-stable/dig/fairgraph

[3] https://github.com/pyg-team/pytorch_geometric

as:

$$\left|\mathbb{P}(\hat{Y}=1 \mid D=0)-\mathbb{P}(\hat{Y}=1 \mid D=1)\right|$$

and EO as:

$$\left|\mathbb{P}(\hat{Y}=1 \mid D=0, Y=1)-\mathbb{P}(\hat{Y}=1 \mid D=1, Y=1)\right|$$

where Y is a ground-truth label, $\hat{Y}$ is a prediction, and in the case of link prediction, D denotes the dyadic group to which the link belongs. These definitions extend to multiple dyadic groups ($|D|>2$), which is the case for subgroup dyadic analysis. As our final metrics, we define the Demographic Parity difference (ΔDP) as the selection rate gap and the Equal Opportunity difference (ΔEO) as the largest discrepancy in true positive rates (TPR) and false positive rates (FPR) across groups identified by D:

$$\begin{aligned}
\Delta\mathrm{DP} &= \max_d \mathbb{P}(\hat{Y}|D=d)-\min_d \mathbb{P}(\hat{Y}|D=d),\\
\Delta\mathrm{TPR} &= \max_d \mathbb{P}(\hat{Y}=1|D=d, Y=1)-\min_d \mathbb{P}(\hat{Y}=1|D=d, Y=1),\\
\Delta\mathrm{FPR} &= \max_d \mathbb{P}(\hat{Y}=1|D=d, Y=0)-\min_d \mathbb{P}(\hat{Y}=1|D=d, Y=0),\\
\Delta\mathrm{EO} &= \max(\Delta\mathrm{TPR}, \Delta\mathrm{FPR}).
\end{aligned}$$

Link prediction performance is measured using accuracy and the Area Under the ROC Curve (AUC) metrics, representing the trade off between true and false positives.

3.5 Computational requirements

All of our experiments are conducted on a high-performance computing (HPC) cluster, that features NVIDIA A100 GPUs, divided into four partitions with a combined memory of 40 GB. For a detailed overview of the GPU hours required for each experiment, see Table 9 in Appendix A.3. In total, 32 of GPU hours were necessary to complete all experiments.

4 Results

4.1 Results reproducing original paper

Claim 1: To verify Claim 1, we perform node classification on the NBA, Pokec-n, and Pokec-z datasets. Table 2 presents a comparison of the results reported by the original authors with those obtained by us through the replication of the experimental setup, as described in Section 3.4. Consistent with the original study, our results are derived by selecting the best outcome from the grid search procedure. We observe that the results for the NBA dataset are similar to those reported by the authors. However, for the Pokec datasets, there is a noticeable difference in accuracy of about 4 to 7 percentage points with similar fairness values. After seeking clarification from the authors and verifying certain default hyperparameters, we conclude that we cannot fully reproduce the results reported in the original paper for the Pokec datasets. Considering that the code provided by the DIG library diverges from what the original authors used, our current hypothesis—following communication with the authors—is that this discrepancy could be due to a bug in the library's code, particularly in the batch functionality necessary for the Pokec datasets. Moreover, this issue might also stems from the observed training instability, where a wide range of hyperparameter settings yield similar outcomes, or where similar configurations lead to large differences.

Table 2: Comparison of the results reported by the original authors with those obtained by us

	NBA			Pokec-n			Pokec-z		
Methods	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓
Original paper	69.36 ± 0.45	2.56 ± 0.41	**4.64 ± 0.17**	**67.43 ± 0.25**	**2.02 ± 0.40**	1.62 ±0.47	**68.17 ±0.08**	**2.10 ± 0.17**	**2.76 ± 0.19**
Our gridsearch	**70.61 ± 0.23**	**1.49 ± 1.06**	4.88 ± 0.59	62.04 ± 0.20	2.37 ± 0.79	**1.22 ± 0.92**	61.97 ± 0.22	3.09 ± 0.36	4.73 ± 0.33

Table 3: Comparisons among different components in the augmentation model.

	NBA			Pokec-n			Pokec-z		
Methods	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓	ACC ↑	Δ_{DP} ↓	Δ_{EO} ↓
Our gridsearch	70.61 ± 0.23	**1.49 ± 1.06**	**4.88 ± 0.59**	62.04 ± 0.20	2.37 ± 0.79	1.22 ± 0.92	61.97 ± 0.22	**3.09 ± 0.36**	**4.73 ± 0.33**
Our Graphair w/o FM	70.89 ± 0.30	7.61 ± 0.35	21.82 ± 0.58	61.78 ± 0.34	**0.97 ± 0.45**	**0.54 ± 0.25**	62.91 ± 0.11	4.08 ± 0.72	5.15 ± 0.31
Our Graphair w/o EP	**73.42 ± 0.38**	5.44 ± 1.10	10.39 ± 2.51	**68.26 ± 0.14**	2.46 ± 0.07	1.82 ± 0.31	**69.18 ± 0.08**	6.42 ± 0.72	6.77 ± 0.63

Claim 2: Table 3 demonstrates that Claim 2 is fully substantiated for both the NBA and Pokec-z datasets. When comparing our results with those obtained without feature masking (FM) or edge perturbation (EP), we observe clear increases in both fairness metrics and accuracy, employing the same hyperparameters as those used for the results presented in Table 2. However, for the Pokec-n dataset, we are unable to fully replicate the results. Our extensive tests, covering a wide range of hyperparameter combinations from our grid search (all yielding results comparable to those noted in Table 2), consistently shows that removing FM results in better fairness metrics while maintaining similar levels of accuracy. Conversely, removing EP leads to significantly improved accuracy with roughly equivalent fairness metrics. This notable discrepancy warrants further investigation into this effect and the specific hyperparameters associated with this behavior.

Claim 3: The plots in Figure 1 clearly illustrate that the homophily values for the augmented graph are lower than those for the original graph. Furthermore, the average homophily values are nearly identical to those reported by the authors. These results support the authors' claim that Graphair automatically generates new graphs with a fairer node topology. The plots in Figure 2 reveal that, for all three datasets, the features with the highest Spearman correlation to the sensitive feature generally exhibit lower values in the fair view. These findings lend support to the authors' claim that Graphair produces features that are fairer. We can therefore conclude that Claim 3 is fully reproducible.

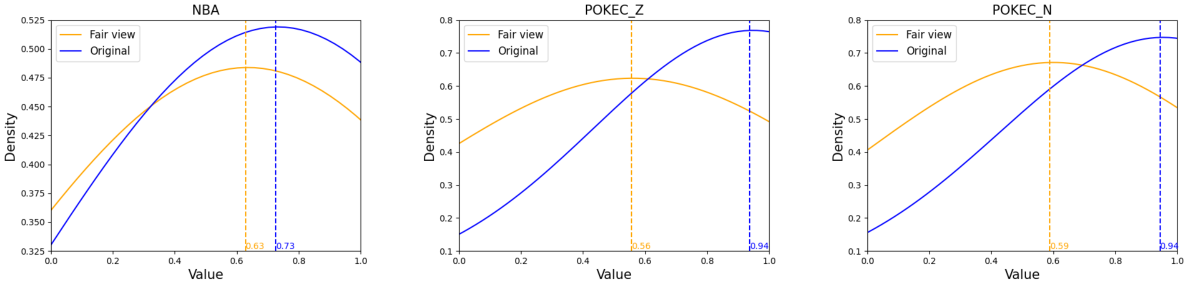

Figure 1: Node sensitive homophily distributions in the original and the fair graph data.

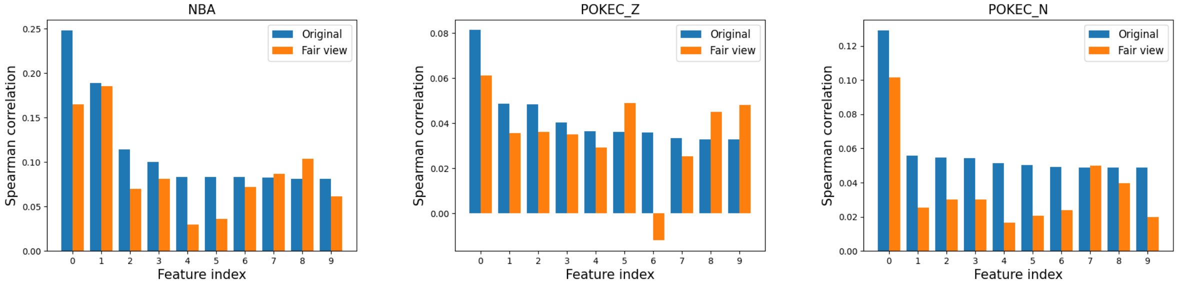

Figure 2: Spearman correlation between node features and the sensitive attribute in the original and the fair graph data.

4.2 Results beyond original paper

We find that Claim 1 can be generalized to link prediction, as evidenced by the results presented in Tables 4, 5, and 6. Details of the citation datasets utilized for this purpose are depicted in Table 1. The results demonstrate the clear superior performance of Graphair across all fairness metrics for the Citeseer and Cora datasets. Given that the accuracy and AUC scores are not significantly compromised, we assert that Claim 1 can be generalized to these datasets. Although the results for the Pubmed dataset do not show improved scores for the DP and EO metrics for the mixed-dyadic group criteria, they do exhibit far superior performance for the subgroup criteria. It is particularly noteworthy how much lower the subgroup fairness metrics are on all datasets compared to those reported in the FairDrop paper (Spinelli et al., 2021). While Spinelli et al. (2021) suggest that their higher scores are likely attributable to the datasets themselves, our results contest this claim. Overall, these results allow us to confidently state that the embeddings generated by Graphair lead to significantly less bias being inherited by the subsequent classifier, as evidenced by the substantially lower fairness scores in the link prediction task on a variety of datasets.

Table 4: Link Prediction on Citeseer

Method	Accuracy ↑	AUC ↑	ΔDP$_m$ ↓	ΔEO$_m$ ↓	ΔDP$_s$ ↓	ΔEO$_s$ ↓
FairAdj$_{r=5}$	78.5 ± 2.2	86.7 ± 2.2	39.2 ± 3.2	19.0 ± 3.9	18.2 ± 5.8	47.6 ± 8.8
FairAdj$_{r=20}$	74.4 ± 2.5	82.5 ± 2.7	31.0 ± 3.1	15.6 ± 3.0	19.7 ± 6.9	43.1 ± 7.4
GCN+FairDrop	**79.2** ± 1.4	**88.4** ± 1.4	42.6 ± 2.5	26.5 ± 4.2	17.6 ± 5.5	64.3 ± 9.5
GAT+FairDrop	78.2 ± 1.1	87.1 ± 1.1	42.9 ± 2.2	28.3 ± 4.3	25.9 ± 5.2	73.4 ± 9.1
Graphair	70.2 ± 0.5	77.9 ± 0.3	**28.3** ± 0.3	**4.5** ± 1.0	**5.4** ± 2.5	**21.1** ± 3.3

Table 5: Link Prediction on Cora

Method	Accuracy ↑	AUC ↑	ΔDP$_m$ ↓	ΔEO$_m$ ↓	ΔDP$_s$ ↓	ΔEO$_s$ ↓
FairAdj$_{r=5}$	75.9 ± 1.6	83.0 ± 2.2	40.7 ± 4.1	20.9 ± 4.3	83.8 ± 4.9	98.3 ± 7.2
FairAdj$_{r=20}$	71.8 ± 1.6	79.0 ± 1.9	32.3 ± 2.8	15.8 ± 4.3	78.34 ± 6.8	98.3 ± 7.2
GCN+FairDrop	**82.4** ± 0.9	**90.1** ± 0.7	52.9 ± 2.5	31.0 ± 4.9	89.4 ± 3.4	100.0 ± 0.0
GAT+FairDrop	79.2 ± 1.2	87.8 ± 1.0	48.9 ± 2.8	31.9 ± 4.3	94.5 ± 2.0	100.0 ± 0.0
Graphair	71.3 ± 0.1	78.8 ± 0.1	**27.3** ± 0.2	**2.6** ± 0.3	**38.9** ± 0.7	**8.9** ± 0.7

Table 6: Link Prediction on PubMed

Method	Accuracy ↑	AUC ↑	ΔDP$_m$ ↓	ΔEO$_m$ ↓	ΔDP$_s$ ↓	ΔEO$_s$ ↓
FairAdj$_{r=5}$	75.5 ± 2.5	84.1 ± 0.7	23.2 ± 4.1	15.9 ± 4.7	53.4 ± 1.9	43.2 ± 9.5
FairAdj$_{r=20}$	73.8 ± 2.4	82.1 ± 0.8	**22.9** ± 4.2	15.9 ± 4.0	52.5 ± 9.7	43.5 ± 9.8
GCN+FairDrop	**88.4** ± 0.4	**94.8** ± 0.2	42.5 ± 0.5	**12.2** ± 0.7	55.7 ± 1.5	26.6 ± 2.6
GAT+FairDrop	79.0 ± 0.8	87.6 ± 0.7	37.4 ± 0.9	19.7 ± 1.1	56.8 ± 2.1	47.3 ± 4.1
Graphair	78.6 ± 0.2	82.0 ± 2.9	24.8 ± 0.4	14.6 ± 0.7	**29.6** ± 2.4	**18.8** ± 2.4

5 Discussion

Revisiting the three claims in our study, we discover that Claims 1 and 2 are partially reproducible, whereas Claim 3 is fully reproducible. The discrepancies encountered in Claim 1, particularly the performance discrepancies on the Pokec datasets which did not meet the reported standards, are likely due to two main issues: unstable training, as evidenced by our grid search results, and a bug within the code's batching functionality, as acknowledged by the original authors. This bug specifically influences the performance on the Pokec datasets. Furthermore, we extend the scope of Claim 1 to assess its applicability to link prediction tasks across different datasets. Here, we observe that our approach outperforms state-of-the-art models, affirming Claim 1's generalizability to the link prediction domain.

The partial reproducibility of Claim 2 seems to stem from the same factors that affected Claim 1. However, it is important to note that we are able to fully reproduce the claim for the NBA and Pokec-z datasets. Interestingly, the outcomes for the Pokec-n dataset are contrary to expectations, suggesting that the model actually improved with the changes instead of worsening.

Claim 3, on the other hand, is fully reproducible. The evidence is clear in both the topology and feature aspects, with homophily values and Spearman correlations demonstrating enhanced results in the augmented graph data compared to the original datasets. This solidifies the claim that Graphair successfully produces fairer graph data by improving both the graph structure and feature correlations.

5.1 What was easy

The comprehensiveness and clarity of the code within the full module significantly facilitated ease of use and implementation. The original paper provided a clear outline of the experiments, enabling a straightforward process to determine the necessary components for reproducing the study's claims. Furthermore, the clarity with which the original paper was written made it relatively easy to fully understand the approach taken. This clarity also simplified the process of making modifications for extending the study to include an additional downstream task, as it was clear which parts of the code were responsible for specific functions and where changes needed to be made.

5.2 What was difficult

Challenges with the consistency and reproducibility of results from hyperparameter grid searches necessitated seeking clarification from the authors, especially since certain default hyperparameters set in the code deviated from those detailed in the original paper. Despite the clarity of the paper and the understandability of the model code, achieving complete reproducibility was not trivial due to these initial discrepancies. These were further enhanced by some commits whose purposes have been unclear since the publication of the original work. Additionally, the authors acknowledged a bug in the provided code as part of the library, specifically in the minibatching code, which likely contributed to the observed deviations leading to conclusions of partial reproducibility for some claims. Finally, reproducing Claim 3 for the Pokec datasets proved non-trivial due to the large memory requirements for processing the full graph, necessitating solutions to acquire experimental results. The GPU hours required to train Graphair on larger datasets also limited our ability to expand on our experimental setup without running into time constraints.

5.3 Communication with original authors

We initiated contact with one of the authors, Hongyi Ling, via email to seek clarification on the hyperparameters used, the dataset, and certain aspects of the code. For instance, we obtained the missing method of mini-batching from their GitHub repository, and the files containing the synthetic graph dataset, which were used to assess the model's scalability. The authors responded promptly to our emails and offered valuable responses.

References

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

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.

Enyan Dai and Suhang Wang. Learning fair graph neural networks with limited and private sensitive attribute information. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674*, 2017.

William L Hamilton. *Graph representation learning*. Morgan & Claypool Publishers, 2020.

Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes. *Advances in Neural Information Processing Systems*, 35:8291–8303, 2022a.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, Qingquan Song, Jundong Li, and Xia Hu. Geometric graph representation learning via maximizing rate reduction. In *Proceedings of the ACM Web Conference 2022*, pp. 1226–1237, 2022b.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

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*, 2022.

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.

Youzhi Luo, Michael McThrow, Wing Yee Au, Tao Komikado, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Automated data augmentations for graph classification. *arXiv preprint arXiv:2202.13248*, 2022.

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

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.

Tong Zhao, Wei Jin, Yozen Liu, Yingheng Wang, Gang Liu, Stephan Günnemann, Neil Shah, and Meng Jiang. Graph data augmentation for graph machine learning: A survey. *arXiv preprint arXiv:2202.08871*, 2022.

A Appendix

A.1 Overview of the Graphair model

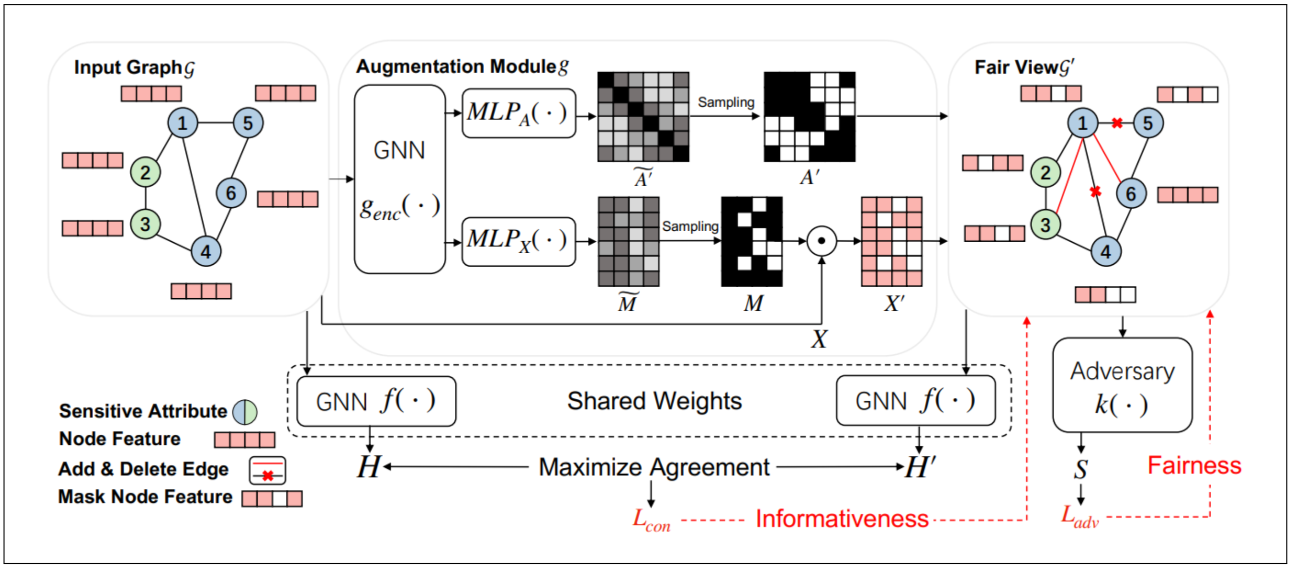

Figure 3: Overview of the Graphair framework (Ling et al., 2022)

A.2 List of all hyperparameters

Table 7: Overview of all hyperparameters of the Graphair components, adverserial model k, representation encoder f and augmentation model g. *For k_{nclass} on link prediction, this value is based on the dataset used between Citeseer, Cora and PubMed.

Hyperparameter	Node Classification	Link Prediction
k_{hidden}	64	64
k_{nclass}	1	{6, 7, 3}*
f_{hidden}	64	64
$\text{f}_{dropout}$	0.1	0.1
f_{nlayer}	2	3
f_{out_feats}	64	64
g_{hidden}	64	64
$\text{g}_{temperature}$	1	1

Table 8: Overview of hyperparameters for the Graphair model m and the evaluation classifier c on all datasets.

Hyperparameter	NBA	Pokec-n	Pokec-z	Citeseer	Cora	PubMed
α	1.0	0.1	10.0	0.1	10.0	10.0
β	0.1	1.0	10.0	0.1	10.0	10.0
γ	0.1	0.1	0.1	0.1	0.1	0.1
λ	1.0	10.0	10.0	1.0	10.0	0.1
c_{hidden}	128	128	128	128	128	128
$\mathrm{c}_{learning\_rate}$	1e-3	1e-3	1e-3	5e-3	5e-3	5e-3
$\mathrm{c}_{weight\_decay}$	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5
$\mathrm{m}_{learning\_rate}$	1e-4	1e-4	1e-4	1e-4	1e-4	1e-4
$\mathrm{m}_{weight\_decay}$	1e-5	1e-5	1e-5	1e-5	1e-5	1e-5

A.3 GPU Run hours

Table 9: Computational Requirement Overview

Name of the experiment	GPU Hour (Hour)	Max GPU Memory Usage (GB)
Claim 1	0.5	3.70
Claim 2	1	3.67
Claim 3	0.5	3.70
Link Prediction	1	3.70
Grid Search 3.3	30	3.70

GPU hour is measured by the amount of time each experiment script needed from the start to the end. Maximum GPU memory usage is determined by *max_memory_allocated* method from the Pytorch library.