A Reproducibility Study: Revisiting fairGNN-WOD and Extending to Multiple Demographics

Anonymous authors
Paper under double-blind review

Abstract

Most existing fair GNN methods assume full or partial access to sensitive demographic attributes during training, which may be unrealistic or infeasible in many real-world settings. FairGNN-WOD (Wang et al., 2025) is a recently proposed two-stage framework that infers sensitive demographic attributes and enforces fairness across the inferred groups. In this paper, we reproduce fairGNN-WOD, provide a more detailed explanation of the framework, and compare its performance with established baselines. We investigate empirical claims made by the authors, which indicate that their fairGNN-WOD leads to improvements in fairness while retaining utility performance. Their framework is tested on the same datasets used in the original paper, along with an additional dataset. Based on our results, we find that there is partial support for the claims made by the authors. We further investigate their model through ablation studies, which give a deeper understanding of the performance of the demographic inference stage. Finally, we extend the method to also be able to account for multiple demographic attributes. We test this extension on the same datasets, and achieve a substantial increase in fairness against the baseline models. The code is publicly available.[1]

1 Introduction

Graph structures are often present in datasets and are already effectively utilized through Graph Neural Networks (GNNs), such as for social networks (Sharma et al., 2024) and financial settings (Wang et al., 2022). However, even though GNNs might perform well for tasks such as loan default prediction and fraud detection, they often achieve this by exploiting biases present in the graph structure, such as homophily or skewed neighborhood connectivity, potentially at the cost of fairness (Chen et al., 2024). Prior work has shown that these structural properties of graphs can lead to biased node representations (Kose & Shen, 2022), and that such biases can further propagate across models, for instance through knowledge distillation where student models inherit the biases of their teachers (Dong et al., 2023). Therefore, it is essential that these graph learning algorithms employ fairness and do not discriminate toward certain demographic subgroups (Dai & Wang, 2021).

Achieving graph fairness is complicated due to three main challenges; (i) preventing label information leakage is a difficult task when trying to infer potential missing demographics, (ii) models exploiting the subgroup identification step to improve fairness in an artificial manner, (iii) fairness constraints leading to performance losses due to the removal of task-related demographic information. These challenges are tackled in the paper by Wang et al. (2025) with the introduction of Fair Graph Neural Network Without Demographics (fairGNN-WOD). The authors introduce a two-stage framework to treat different demographic groups fairly, while also allowing for the lack of complete demographics. Essentially, a model should be fair toward demographics, regardless of whether the attribute is available to the model. For example, when applying a model in hiring processes, it is desirable that the employment is fair toward gender. Nonetheless, applicants may choose to withhold their information. A model should then still be able to fairly choose new employees with regards to their gender, even though it does not know the applicants' gender explicitly.

† All authors contributed equally.
[1] https://anonymous.4open.science/r/fairGNN-WOD-CD51
https://anonymous.4open.science/r/graph-fairness-data-4321

The framework uses Bayesian Variational Autoencoders (VAEs) to infer missing demographic information in data, where the inferred demographics are then utilized in the second stage as additional information to help GNNs learn fair representations for downstream tasks. The fairGNN-WOD is built on the insights from the authors' bias analysis on how demographic information can propagate through node representations and lead to disparate outcomes across different subgroups. The model is then tested on three public datasets; the Credit dataset (Yeh & Lien, 2009), and the Pokec-z and Pokec-n datasets (Takac & Zabovsky, 2012). Benchmarking the proposed framework against different baseline models lead the authors to claim that their method is effective in mitigating unfairness while maintaining comparable performance.

This paper studies the reproducibility of the results reported by Wang et al. (2025), assessing whether their framework achieves improved fairness while maintaining comparable utility. We reconstruct their model, evaluate it on the original datasets alongside the same baselines, and extend the analysis to the Recidivism dataset (Jordan & Freiburger, 2015) to test the generalization of the reported performance trends. In addition, we provide a fully specified and reproducible implementation of fairGNN-WOD, clarifying missing details and finding only partial support for the original claims. To further analyze the method, we introduce two ablation studies examining the role of demographic inference and the underlying independence assumptions, and extend the framework to multiple demographic attributes (fairGNN-WMD), demonstrating improved fairness across datasets.

2 Scope of Reproducibility

This study focuses on the empirical claims made by Wang et al. (2025) regarding the proposed fairGNN-WOD framework. They compare their method against six different baselines, where two are vanilla graph models not incorporating fairness and four models are methods enhancing fairness but not built towards graph data specifically. The following claims are made based on their results and will be verified in this study:

- *Claim 1*: fairGNN-WOD shows a significant improvement in fairness over the vanilla graph methods.
- *Claim 2*: fairGNN-WOD performs better with regards to fairness than the fairness methods adapted from non-graph domains.
- *Claim 3*: fairGNN-WOD outperforms the baselines regarding fairness in all cases and surpasses in utility performance in most cases.

This study aims to provide further insights with the intention of filling missing gaps of the original paper. Specifically, we consider two ablation studies. Firstly, their framework includes a method to infer missing demographics, for which they omitted to show that the inferring of missing demographics actually helps the model improve in fairness. Therefore, we conduct an ablation study to understand the contribution of this mechanism. Secondly, to infer demographics, the authors rely on an independence assumption, for which no justification is given. Hence, we relax this assumption and present how performances differ.

Finally, the authors only introduce and apply their framework in the situation where a single demographic attribute is considered. In real-life scenarios, this simplification does not hold, as often more sensitive attributes are present. Thus, we also present how this model can be extended to multiple demographics and test it on the datasets using two and three demographic attributes.

3 Methodology

In the original paper, key details were missing for a fair portion of the framework. Therefore, this study needed to adapt the main ideas of the model and rebuild the entire framework, not specified in itself by Wang et al. (2025). Hence, this description of the model will go beyond the one given by the original authors and aims to recover the original implementation.

The complete framework consists of two stages. The first stage, covered in subsection 3.1, is meant to account for scenarios where complete demographics is not available through latent representation learning. It leverages observed prior knowledge to infer the missing demographic information, while excluding target

label information to ensure fairness. The second stage, covered in subsection 3.2, utilizes the inferred demographics as proxies to mitigate bias in graph learning. Moreover, the framework is extended to allow for multiple demographic attributes, with the required modifications described in subsection 3.3.

For the remainder of the paper, similar notation from the original authors is adopted. $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$ denotes an undirected graph containing attributes for nodes, where $|\mathcal{V}| = n$ denotes the number of nodes. The i-th row of the feature matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a d-dimensional feature vector for node i. $\mathbf{A} \in \{0,1\}^{n \times n}$ denotes the adjacency matrix, where $A_{i,j} = 1$ if there exists an edge $(i,j) \in \mathcal{E}$ between node i and j, and $A_{i,j} = 0$ otherwise. The set of neighbors for node i is denoted as $\mathcal{N}(i)$. Demographics are assumed to be binary variables. Moreover, for the reproduction of the original paper, it is assumed that for a single dataset there is only one sensitive attribute. $\mathbf{S} \in \{0,1\}^{n \times 1}$ denotes the binary demographic, with s_i being the demographic value for node i. For the purpose of fairness, $\mathbf{S}_d = \{i \in \mathcal{V} | s_i = 0\}$ represents the deprived group and $\mathbf{S}_f = \{i \in \mathcal{V} | s_i = 1\}$ represents the favored group. The final task for this model is node classification and target labels are assumed to be binary, i.e. $\mathbf{Y} \in \{0,1\}^{n \times 1}$. The ground truth label of node i is denoted by y_i, and its predicted label as $\hat{y}_i$.

3.1 Stage 1 - Inferring Missing Demographics

The main idea of the first stage is to learn the underlying data generation process through an additional latent representation $\mathbf{Z}$. This latent space is meant to encapsulate as much information about $\mathbf{S}$ as possible such that the graph data can be reconstructed. If we can accurately recover the joint distribution $P(\mathbf{Z}, \mathbf{S}, \mathbf{A}, \mathbf{X})$, it is possible to recover the missing demographic information (Louizos et al., 2017). For this reason, Bayesian inference approximation is used, where the joint probability is factorized as

$$P(\mathbf{Z}, \mathbf{S}, \mathbf{A}, \mathbf{X}) = P(\mathbf{Z}) \cdot P(\mathbf{S}|\mathbf{Z}) \cdot P(\mathbf{A}|\mathbf{S}) \cdot P(\mathbf{X}|\mathbf{A}, \mathbf{S}), \tag{1}$$

under the assumption that $\mathbf{Z}$ is independent from $\mathbf{A}$ given $\mathbf{S}$, and from $\mathbf{X}$ given $\mathbf{A}$ and $\mathbf{S}$. This assumption is made implicitly by the authors without any justification, and will be further investigated by testing the model without the use of the assumption. $P(\mathbf{Z})$ represents the prior over the latent distribution of $\mathbf{Z}$. $P(\mathbf{A}|\mathbf{S})$ and $P(\mathbf{X}|\mathbf{A}, \mathbf{S})$ denote the decoders to obtain the reconstructions $\hat{\mathbf{A}}$ and $\hat{\mathbf{X}}$ respectively, without the use of $\mathbf{Z}$. The term $P(\mathbf{S}|\mathbf{Z})$ is not discussed by Wang et al. (2025), leading to ambiguity regarding the meaning and implementation of the term. To figure out what this term could represent and how it could have been represented, we first need to investigate the loss function thoroughly.

Part of this loss function stems from the type of framework used. The first stage combines ideas of variational graph autoencoders (VGAEs) (Kipf & Welling, 2016) and deep hierarchical VAEs (Vahdat & Kautz, 2020). Two variational distributions are introduced, $Q(\mathbf{Z}|\mathbf{X}, \mathbf{A})$ and $Q(\mathbf{S}|\mathbf{Z})$, which are approximated by two sequential encoders; $q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})$ and $q_\psi(\mathbf{S}|\mathbf{Z})$. They are used to maximize the Evidence Lower Bound (ELBO) of the marginal data likelihood $P(\mathbf{X}, \mathbf{A})$ (Kingma & Welling, 2013). Only the reconstruction of $\mathbf{X}$ and $\mathbf{A}$ is important as we treat the demographics as a latent variable. The derivation of the ELBO can be found in appendix A and leads to

$$\begin{aligned} \log P(\mathbf{X}, \mathbf{A}) \geq\ & \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z}} \left[\log P(\mathbf{A}|\mathbf{S}) + \log P(\mathbf{X}|\mathbf{A}, \mathbf{S}) \right] \right] \\ & - D_{KL} \left(q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \mid\mid P(\mathbf{Z}) \right) \\ & - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})} \left[D_{KL} \left(q_\psi(\mathbf{S}|\mathbf{Z}) \mid\mid P(\mathbf{S}|\mathbf{Z}) \right) \right]. \end{aligned} \tag{2}$$

Note that the ELBO is differently presented here than by the original authors. That said, from this perspective, it becomes evident that the term $P(\mathbf{S}|\mathbf{Z})$ acts as a conditional prior on $q_\psi(\mathbf{S}|\mathbf{Z})$. Yet, there is no guarantee that this is also how the authors interpreted it. Hence, the following implementation regarding this term might deviate from theirs.

Adopting this perspective clarifies different parts of their framework. The model starts with an encoder that models the latent $\mathbf{Z}$ from $\mathbf{X}$ and $\mathbf{A}$ ($q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})$). This encoder is modeled as a graph convolution network (GCN) (Kipf, 2016) that takes as input the graph data and maps them together with two separate final linear layers to a mean and logarithmic variance. These two outputs define the Gaussian distribution from which $\mathbf{Z}$ can be sampled. The prior $P(\mathbf{Z})$ is modeled as a standard Gaussian, influencing the posterior through the

Kullback-Leibler (KL) divergence term. Next, another encoder is applied that maps the sampled $\mathbf{Z}$ to the demographic latent $\mathbf{S}$ ($q_\psi(\mathbf{S}|\mathbf{Z})$).

The variable $\mathbf{S}$ is binary and hence cannot be modeled as a Gaussian, per usual in VAEs. Thus, instead the latent distribution for $\mathbf{S}$ is chosen to be a Bernoulli, as in Sønderby et al. (2016). This encoder takes in the sampled $\mathbf{Z}$ and maps it through a multilayer perceptron (MLP) to logits $f(\mathbf{Z}) \in \mathbb{R}^{n\times 1}$, which are passed through the sigmoid function. Then $\mathbf{S}$ can be sampled from the Bernoulli distribution, i.e. $\mathbf{S} \sim \text{Bernoulli}(\sigma(f(\mathbf{Z})))$. The resulting Bernoulli is regularized by the conditional prior $P(\mathbf{S}|\mathbf{Z})$. As this prior is dependent on $\mathbf{Z}$, this is also modeled as a (less expressive) MLP. These two neural networks will learn different parameters, as the conditional prior is only updated through the KL divergence loss, while the encoder is also updated through the reconstruction loss.

Sampling $\mathbf{S}$ from the Bernoulli distribution leads to non-differentiability. Therefore, during training, sampling will be done through the relaxed Bernoulli distribution (Maddison et al., 2016). Specifically, the reparameterization trick is applied to sample the demographics from this distribution. This allows for the model to be differentiable, with the sampling yielding a value between 0 and 1 rather than a binary value. Note that these details regarding the implementation of the encoders are left out in the original paper, and hence are mathematically reformulated based on how the demographics and ELBO behave.

After sampling $\mathbf{S}$, two decoders are applied to reconstruct $\mathbf{X}$ and $\mathbf{A}$. Before applying the decoders, the latent $\mathbf{S}$ is embedded through an MLP into a higher dimensional space. Then, the decoder $P(\mathbf{A}|\mathbf{S})$, for each pair of nodes, concatenates the embeddings and maps them through an MLP to a score. For the decoder $P(\mathbf{X}|\mathbf{A},\mathbf{S})$, the original adjacency matrix is utilized together with the sampled $\mathbf{S}$, as they are passed through a GCN, yielding the predictions for the feature matrix. This decoder is implemented as a regression, regardless of whether the variables are discrete. The reason for this is because the overall model is not so concerned with the quality of the reconstructions for $\mathbf{X}$ and $\mathbf{A}$. These are only used to train the model to learn an efficient latent space for the demographics. Again here, the implementation of the decoders are not mentioned by the original authors. An overview of the first stage is given in Figure 1.

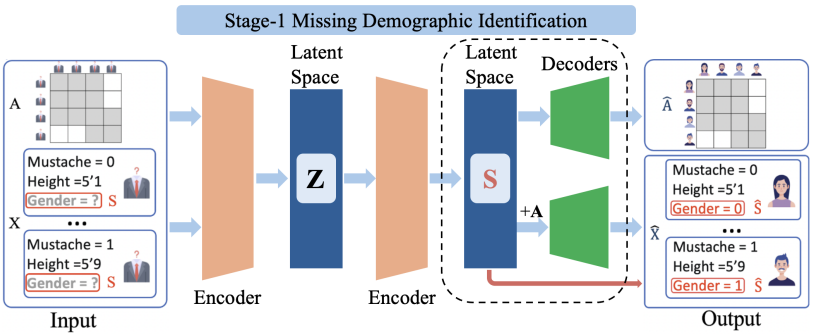

Figure 1: Overview of the first stage of fairGNN-WOD. The dotted box shows our additional detailing of the model, namely $\mathbf{S}$ being sampled from a Bernoulli distribution and explicit use of two decoders: $P(\mathbf{A}|\mathbf{S})$ and $P(\mathbf{X}|\mathbf{A},\mathbf{S})$.

To further ensure a fair inference of $\mathbf{S}$, a penalty term is added to the loss, incentivizing independence from $\mathbf{Y}$, through $\mathbf{Z}$. This penalty term is chosen to be the Hirschfeld-Gebelein-Rényi (HGR) maximal correlation that quantifies a linear or non-linear dependence between the latent $\mathbf{Z}$ and the outcome $\mathbf{Y}$. The definition is given in Appendix B (Lee, 2021).

A value of zero for $\text{HGR}(\mathbf{Z},\mathbf{Y})$ indicates independence between the latent variable and the outcome. The penalty term and the negative ELBO form the objective function to be minimized for the first stage:

$$\mathcal{L}_1 = -\text{ELBO} + \lambda \cdot \text{HGR}(\mathbf{Z},\mathbf{Y}). \tag{3}$$

The objective includes a hyperparameter λ balancing maximization of the ELBO and minimization of correlation. To approximate the HGR maximal correlation term, two MLPs denoted by f and g are applied

which map the respective variable $\mathbf{Z}$ and $\mathbf{Y}$ to a hidden dimension. The two embeddings are multiplied with each other after normalization, where the mean is taken to yield the approximated HGR penalty.

For training the first stage, using the Adam optimizer (Kingma, 2014), the interconnected neural networks f and g are first updated by maximizing $\mathbb{E}[f(\mathbf{Z})g(\mathbf{Y})]$ for a few steps, yielding $\widehat{\text{HGR}}(\mathbf{Z}, \mathbf{Y})$ used for the objective. Then, again using the Adam optimizer, the complete first stage can be trained on the objective. After training the model, the first stage is used for inference of the demographics. For inference, we can sample from the Bernoulli distribution. These inferred demographics $\widetilde{\mathbf{S}}$ are passed onto stage two of fairGNN-WOD, and together with $\mathbf{A}$ and $\mathbf{X}$ form the necessary inputs for the classification task.

3.2 Stage 2 - Fair Graph Learning

Inferred demographics are used in the second stage of the model to suppress any demographic-relevant information in the embeddings, where we denote the embeddings containing this information by $\mathbf{H_S}$. This should enable the model to make predictions for the classification task without the use of demographic-relevant information, and hence improve fairness. A straightforward solution is to remove $\mathbf{H_S}$ completely, as done in existing fairness-aware GNNs (Chen et al., 2024). However, this approach can lead to performance deterioration, due to the removal of task-related information contained in $\mathbf{H_S}$. Therefore, to maintain performance on the final task, the aim of the second stage is to keep task-related information in $\mathbf{H_S}$ while removing demographic information.

In order to implement this approach, the second stage should first be able to find demographic-relevant information in the node representations. For this purpose, a form of disentangled graph convolution is used, inspired by Ma et al. (2019) and similar to fairSAD by Zhu et al. (2024). The main idea is to learn node embeddings that are disentangled into multiple channels, independent from each other. Each of these channels will represent one specific latent factor in the complete node embeddings, whereas only one of these channels should then be identifiable as the one containing demographic-relevant information. Lemma 1, adapted from Zhu et al. (2024), shows that only one component will contain demographic-relevant information through proper disentangled representation learning. The lemma, together with its proof, can be found in Appendix C. From now on, the number of channels will be denoted as N_c.

For disentangled representation learning, the original authors first introduce an adaptive assigner that gives different weights to the neighbors of a node with respect to each latent channel. This mechanism is implemented as an MLP, denoted by $F_{\boldsymbol{\psi}}$. In particular, for an edge $(i, j) \in \mathcal{E}$, it takes as input the concatenation of the features of the nodes $[\mathbf{x}_i, \mathbf{x}_j]$. Then, for these neighboring nodes, $F_{\boldsymbol{\psi}}$ outputs a vector $\boldsymbol{\psi}_{i,j} \in \mathbb{R}^{N_c}$, representing the importance score of the channels $c = 1, ..., N_c$ influencing the connection between node i and j. Note that at this stage these disentangled channels have not yet been learned, but $\boldsymbol{\psi}_{i,j}$ will affect the learning process of these channels after normalization through the softmax function. Here, $\boldsymbol{\omega}_{i,j} = \text{softmax}(\boldsymbol{\psi}_{i,j})$ denotes the normalized weights, where each element $\omega^c_{i,j} \in \boldsymbol{\omega}_{i,j}$ for $c = 1, ..., N_c$ represents the probability of the connection between node i and j being influenced by the latent channel c.

Next, disentangled graph convolution is applied using these weights to learn the channel representations. For this, first the node features $\mathbf{x}_i \in \mathbb{R}^d$ for each node i are put N_c times through a linear transformation specific to each channel. Each of these linear transformations independently applies a dimensionality reduction to obtain the initial node embeddings $\mathbf{h}^{(0)}_{i,c} \in \mathbb{R}^{d'}$ for each channel c and node i, with $d' < d$. These embeddings are then utilized in disentangled layers for graph convolution, where the embeddings over channels are updated independently of each other. For each layer l, denote the hidden embedding for node i and channel c as $\mathbf{h}^{(l)}_{i,c}$. Then, the operation between the l'th and $(l+1)$'th layer for channel c is defined as

$$\mathbf{h}^{(l+1)}_{i,c} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \omega^c_{i,j} \cdot \mathbf{h}^{(l)}_{j,c} \cdot \mathbf{W}^{c,(l)}\right), \tag{4}$$

where the neighborhood $\mathcal{N}(i)$ includes node i itself, $\mathbf{W}^{c,(l)}$ is a learnable weight matrix for channel c in layer l, and $\sigma(\cdot)$ denotes an activation function. Note that the dimensionality d' is preserved throughout all L

layers. The final hidden node embeddings $\mathbf{h}_{i,c}^{(L)}$ are collected across all nodes per channel to obtain the final channel representations $\mathbf{K}_1, ..., \mathbf{K}_{N_c}$. Here, $\mathbf{K}_c = [\mathbf{h}_{1,c}^{(L)}, ..., \mathbf{h}_{n,c}^{(L)}]$ with $\mathbf{K}_c \in \mathbb{R}^{n \times d'}$.

This mechanism allows the model to learn disentangled node representations across channels. To encourage the model to obtain final disentangled representations, the authors introduce an Independence Constraint based on the Maximum Mean Discrepancy (MMD) empirical estimate by Gretton et al. (2006). In particular, for channels c_1 and c_2, with channel representations $\mathbf{K}_{c_1}$ and $\mathbf{K}_{c_2}$, the MMD estimate is defined as

$$\mathrm{MMD}^2(\mathbf{K}_{c_1}, \mathbf{K}_{c_2}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f(\mathbf{K}_{c_1,i},\ \mathbf{K}_{c_1,j}) + \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f(\mathbf{K}_{c_2,i},\ \mathbf{K}_{c_2,j}) - \frac{2}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} f(\mathbf{K}_{c_1,i},\ \mathbf{K}_{c_2,j}). \quad (5)$$

Here, $f(\cdot,\cdot)$ is a kernel function, and $\mathbf{K}_{c_1,i}$ refers to $\mathbf{h}_{i,c_1}^{(L)}$. This estimate is calculated over every pair (c_1, c_2) with $c_1 < c_2$, and the negative sum forms the Independence Constraint:

$$\mathcal{L}_I = -\sum_{c_1=1}^{N_c}\sum_{c_2=c_1+1}^{N_c} \mathrm{MMD}^2(\mathbf{K}_{c_1}, \mathbf{K}_{c_2}). \quad (6)$$

The minimization of this loss term encourages the disentanglement between channels, leading to the desired hidden embeddings. Through this, it will now be possible to identify one channel containing demographic-relevant information.

To identify this channel, a discriminator is utilized that takes the channel embeddings $\mathbf{K}_c$ as input and predicts the demographic labels for all nodes. Specifically, the discriminator is implemented as an MLP, and outputs a probability for the demographics using the embeddings of channel c, denoted by $\hat{s}_{i,c}$. The inferred demographics $\tilde{s}_i$ from stage 1 serve here as the ground truth. The idea here is that the channel leading to the most accurate predictions through the discriminator should be the one containing the demographic-relevant information. The discriminator is trained simultaneously, and the classification loss for it is defined as

$$\mathcal{L}_D = -\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\sum_{c=1}^{N_c}\left[\widetilde{s}_i \cdot \log(\hat{s}_{i,c}) + (1-\widetilde{s}_i)\cdot\log(1-\hat{s}_{i,c})\right]. \quad (7)$$

The channel containing demographics is chosen by using the predicted probabilities from each channel embedding and calculating the binary cross entropy loss defined above. The channel leading to the lowest loss is chosen as the channel containing demographic-relevant information, denoted by $\hat{c}$. This operation is applied without tracking gradients. Moreover, the embeddings being put into the discriminator are detached so that no gradients from the discriminator can affect the embeddings themselves. This leads to this block being detached from the rest of the model, allowing it to be trained online, while not affecting the updates of weights for the rest of the model.

At this point, we have all the channel embeddings, and know which channel contains the demographic-relevant information. The embeddings from channel $\hat{c}$ will be denoted as $\mathbf{H_S}$ and the embeddings of the other channels are together denoted as $\mathbf{H_X}$, i.e. we split the embeddings into the ones relevant to demographics and the ones relevant to the features. The next step is to process $\mathbf{H_S}$ to enhance fairness. With this aim, a learnable mask is applied on the hidden embeddings to suppress the demographic-relevant information present in $\mathbf{H_S}$, while retaining task-relevant information. Specifically, this mask is implemented as a learnable parameter vector $\mathbf{m} \in \mathbb{R}^{d'}$ and applied as a soft mask on $\mathbf{H_S}$ per node, along the hidden dimensionality. This results in the masked embeddings $\widetilde{\mathbf{H}}_\mathbf{S}$, which can then be used for the downstream task. To ensure that this mask effectively eliminates demographic information from $\mathbf{H_S}$, another loss term is added that penalizes the absolute covariance between the inferred demographics from stage one and final node label prediction of the complete stage one, denoted by $\hat{y}_i$:

$$\mathcal{L}_F = \left|Cov(\tilde{\mathbf{S}}, \hat{\mathbf{y}})\right| = \left|\mathbb{E}\left[(\tilde{\mathbf{S}} - \mathbb{E}(\tilde{\mathbf{S}})\ (\hat{\mathbf{y}} - \mathbb{E}(\hat{\mathbf{y}}))\right]\right|, \quad (8)$$

where $\hat{\mathbf{y}}$ collects all node predictions as a vector.

The masked embeddings $\widetilde{\mathbf{H}}_{\mathbf{S}}$ and the embeddings from other channels $\mathbf{H}_{\mathbf{X}}$ together form the fair representations per node, used for the final classification task. For effective training, a final classification loss on the label predictions for each node is included as

$$\mathcal{L}_P = -\frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}}\left[y_i \cdot \log(\hat{y}_i) + (1-y_i)\cdot\log(1-\hat{y}_i)\right]. \tag{9}$$

Together with the previously introduced loss terms (Eqs. 6, 7, 8, 9), the loss function for the second stage is formed as

$$\mathcal{L}_2 = \mathcal{L}_P + \alpha\mathcal{L}_I + \alpha\mathcal{L}_D + \beta\mathcal{L}_F, \tag{10}$$

where α and β are tunable hyperparameters. The minimization of this function leads to the simultaneous training of the detached discriminator block and the rest of stage two. An overview of the second stage is provided in Figure 2.

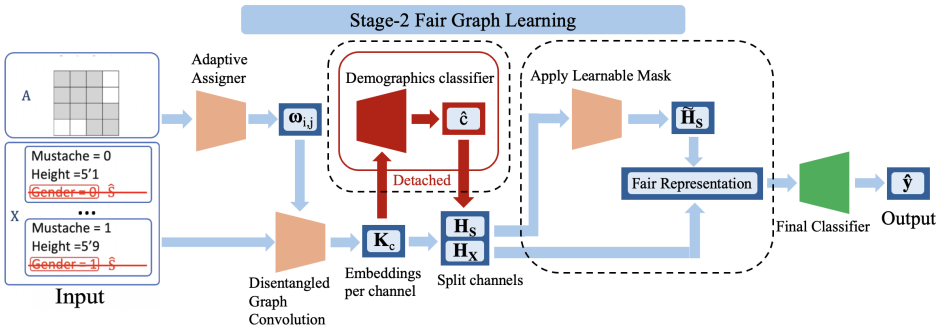

Figure 2: Overview of the second stage of fairGNN-WOD. Note that the inferred demographics are only used for finding the channel containing demographic-relevant information.

The entire fairGNN-WOD is trained in a two-step manner. First, stage one is optimized, after which the weights are frozen. Then, stage two is optimized using the inference of demographics from the frozen first stage.

3.3 Extending to Multiple Demographic Attributes

So far, the framework has been based on having a single demographic attribute. This is, however, often not the case in real-life scenarios. Thus, we extend the framework to account for multiple demographics, while keeping the methods as close to the original as possible. In this section, we will show the necessary changes that need to be made for this extension, but when testing the framework we will only consider the cases where two or three demographic attributes are included. From now on, the framework for multiple demographics will be referred to as fairGNN-WMD (Fair Graph Neural Network Without Multiple Demographics).

The joint distribution for recovering the missing demographic information in the first stage now includes multiple demographic attributes denoted by $\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m$. For the factorization of this joint probability, we now additionally assume independence between the demographic attributes given the latent $\mathbf{Z}$. Then

$$P(\mathbf{Z}, \mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m, \mathbf{A}, \mathbf{X}) = P(\mathbf{Z}) \cdot P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z}) \cdot P(\mathbf{A}|\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m) \cdot P(\mathbf{X}|\mathbf{A}, \mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m), \tag{11}$$

where $P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z}) = P(\mathbf{S}_1|\mathbf{Z}) \cdot P(\mathbf{S}_2|\mathbf{Z}) \cdot \ldots \cdot P(\mathbf{S}_m|\mathbf{Z})$ through the independence assumption. Therefore, we now also introduce m variational distributions $Q(\mathbf{S}_1|\mathbf{Z}), Q(\mathbf{S}_2|\mathbf{Z}), ..., Q(\mathbf{S}_m|\mathbf{Z})$, which are approximated by one encoder $q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})$. This encoder will now output m different logits, each passed through the sigmoid function. Each resulting probability represents the parameter of the Bernoulli distribution for each demographic attribute. Due to the independence between attributes through $\mathbf{Z}$, we can sample the demographics attributes independently from each resulting Bernoulli distribution. Then, each sampled demographic attribute is passed on to the decoders of the adjacency and feature matrix as before, where now the number of inputs to the decoders is m.

The ELBO that is included for the loss function of the first stage also changes based on the inclusion of more sensitive attributes. We still reconstruct $\mathbf{X}$ and $\mathbf{A}$, and hence maximize the ELBO of the marginal data likelihood $P(\mathbf{X}, \mathbf{A})$. Before, the ELBO consisted of three parts; the reconstruction loss, and a KL divergence loss for $\mathbf{Z}$ and $\mathbf{S}$. The first two parts do not change, but the KL divergence for the demographic attributes is now defined for the encoder $q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})$ and the conditional prior $P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})$. The encoder and conditional prior here are modeled in a similar manner, with the only difference being that they now output m probabilities instead of one. The KL divergence term can then be equivalently calculated as

$$\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}\left(q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z}) \mid\mid P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})\right)\right] = \sum_{i=1}^{m} \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}\left(q_\psi(\mathbf{S}_i|\mathbf{Z}) \mid\mid P(\mathbf{S}_i|\mathbf{Z})\right)\right], \tag{12}$$

with $q_\psi(\mathbf{S}_i|\mathbf{Z})$ and $P(\mathbf{S}_i|\mathbf{Z})$ representing the individual probabilities outputted by the neural networks. The complete ELBO and derivation can be found in Appendix A.

Through these changes, the first stage is now able to infer multiple demographics at once. As these inferred demographics are not a direct input for the next stage, nothing changes for the second stage up to the demographic classifier. Before, this classifier was used to identify the channel containing the demographic-relevant information, based on Lemma 1 that at most one channel is able to contain this information. With the same logic, as we now have m demographic attributes, there will be at most m channels containing demographic-relevant information. Therefore, instead of having just one classifier identifying the relevant channel, we have m classifiers, each geared toward identifying one channel containing relevant information regarding one specific attribute. Note that we did not assume full independence between demographics, and hence it is possible that two or more classifiers identify the same channel. For the training of each classifier, we include a classification loss like Equation 7 for each separate demographic attribute $\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m$, denoted by $\mathcal{L}_{D_1}, \mathcal{L}_{D_2}, ..., \mathcal{L}_{D_m}$. The total loss for identifying the channels then becomes $\mathcal{L}_D = \mathcal{L}_{D_1} + \mathcal{L}_{D_2} + \ldots + \mathcal{L}_{D_m}$, and is included in $\mathcal{L}_2$ as before (Equation 10).

As there are now m potential channels containing demographic-relevant information, we also include m separate soft masks per demographic attribute. These masks are the same as defined before. If in the case that one channel is identified multiple times as the relevant channel for some sensitive attributes, then the masks corresponding to these demographics are applied sequentially on that same channel. Each mask is supposed to eliminate any information with regards to some specific sensitive attribute, and hence we include a loss term penalizing the absolute covariance between each demographic attribute and the final node label prediction, as in Equation 8. These loss terms per demographic attributes are denoted by $\mathcal{L}_{F_1}, \mathcal{L}_{F_2}, ..., \mathcal{L}_{F_m}$ and the sum becomes the total loss meant for the training of the masks. This loss, $\mathcal{L}_F = \mathcal{L}_{F_1} + \mathcal{L}_{F_2} + \ldots + \mathcal{L}_{F_m}$, is also included in $\mathcal{L}_2$ as before.

All of these changes to the framework allow for the inclusion of more than one demographic attribute. Any parts of fairGNN-WOD that are omitted here, remain the same for the extended model fairGNN-WMD.

4 Experimental Setup

In this section, a short description for each dataset is given, the baseline models are introduced, and the metrics used to evaluate the models are presented. Hyperparameters, computational requirements, and additional descriptions of the datasets with the used demographic attributes can be found in Appendix D.

4.1 Datasets

The fairGNN-WOD and fairGNN-WMD are tested on the same datasets used in the original paper. This includes three publicly available datasets: the Credit dataset (Yeh & Lien, 2009), and the Pokec-z and Pokec-n datasets (Takac & Zabovsky, 2012). Next to these datasets, this paper also includes the Recidivism dataset (Jordan & Freiburger, 2015). The Recidivism dataset is especially important for fairness-aware prediction because the model's decisions directly affect people's lives, as they can influence judicial outcomes such as whether a defendant is granted bail or remains detained. This introduces ethical considerations that are less

notable in the other datasets. By including this dataset, we strengthen the evaluation of fairGNN-WOD and fairGNN-WMD by testing it in a setting where fairness has a clear societal implication.

The Credit dataset includes information about credit card holders at a bank in Taiwan, the task is to predict default payments in the next month. The Pokec-z and Pokec-n datasets stem from a social network in Slovakia. Nodes represent users, and connections between nodes represent friendships between users. The goal is to predict whether users are employed. The Recidivism dataset includes information on defendants who were released on bail at U.S. state courts. The task is to classify defendants on whether they should be allowed to go on bail.

4.2 Baselines

We compare fairGNN-WOD with the same baselines as in the original paper. These baselines can be categorized into two groups; vanilla graph models that do not take fairness into account, and fairness models that are not yet adapted for graph structured data. The vanilla graph models used are the GCN (Kipf, 2016) and GIN (Xu et al., 2018). The fairness baseline methods are adopted to graph structural data by applying a GCN encoder that provides node representations, which are then used for the fairness approaches. This GCN encoder stems from the implementation of the GCN baseline model, where the parameters are frozen to obtain the node representation. The fairness baseline models include FairKD (Chai et al., 2022), KSMOTE (Yan et al., 2020), FairRF (Zhao et al., 2022), and Reckoner (Ni et al., 2024). For the comparison of fairGNN-WMD, these baseline models are additionally extended to account for multiple demographic attributes.

4.3 Metrics

To evaluate the performance of the different models, we calculate the accuracy and F1-score. For comparing fairness among the different approaches, two common fairness metrics are calculated. Namely, Statistical Parity Differences (SPD) introduced by Dwork et al. (2012) and Equal Opportunity Differences (EOD) by Hardt et al. (2016). These fairness metrics are reported as the absolute value and a value closer to zero indicates better fairness. For comparing fairGNN-WMD with the baselines, we use the same metrics extended to multiple demographics (Chen et al., 2024). The relevant formulas can be found in Appendix D.2.

5 Results

5.1 Reproducing the Original Paper

The results for fairGNN-WOD against the mentioned baselines can be found in Table 1. Compared to the results reported by the original authors, our implementations achieve consistently stronger performance across both utility and fairness metrics. For comparison, the results of the original authors have been included in Appendix G, where a direct side-by-side analysis highlights substantial discrepancies across most baseline methods, suggesting differences in experimental conditions or evaluation protocols. In contrast, the performance of fairGNN-WOD remains relatively consistent, and given our fully specified implementation and open-source code, we argue that our findings are reproducible. However, our results do not fully align with the claims made by Wang et al. (2025).

The authors found that their framework shows a significant improvement in fairness over the vanilla graph methods (*Claim 1*). Looking at our results, we can confidently say that this claim holds for the Credit, Pokec-z and Recidivism dataset. However, on the Pokec-n dataset, fairGNN-WOD struggles and does not manage to achieve better fairness as compared to GCN and GIN. Next, the authors stated that their framework demonstrates better fairness performance as compared to the fairness methods adapted from non-graph domains (*Claim 2*). Again here, fairGNN-WOD shows significant improvement in fairness for the Credit and Recidivism dataset with an average decrease of 80% in SPD and 44% in EOD against the best baseline result. However, it performs worse than all baselines on the Pokec-n dataset, and can not outperform the KSMOTE baseline on the Pokec-z dataset. Though, fairGNN-WOD does outperform all other baseline

Table 1: Results of fairGNN-WOD with baseline methods across the used datasets. For each dataset and each metric, the best result is indicated in bold, whereas the runner-up result is marked with an underline. Results are averaged over three runs with different random seeds.

Metric	Dataset \ Method	GCN	GIN	FairKD	KSMOTE	FairRF	Reckoner	fairGNN-WOD
Accuracy (↑)	Credit	0.7855 ± 0.0038	0.7878 ± 0.0009	0.7823 ± 0.0023	0.6704 ± 0.0345	0.7865 ± 0.0026	0.7873 ± 0.0034	**0.7950 ± 0.0066**
	Pokec-z	0.7129 ± 0.0078	0.6646 ± 0.0170	**0.7224 ± 0.0069**	0.6772 ± 0.0028	0.7178 ± 0.0002	0.7202 ± 0.0058	0.7058 ± 0.0086
	Pokec-n	0.7163 ± 0.0049	0.6356 ± 0.0225	**0.7361 ± 0.0058**	0.7104 ± 0.0080	0.7197 ± 0.0096	0.7314 ± 0.0060	0.7028 ± 0.0090
	Recidivism	0.9394 ± 0.0029	0.9560 ± 0.0018	0.9398 ± 0.0018	0.7728 ± 0.0047	0.9548 ± 0.0008	0.9499 ± 0.0013	**0.9843 ± 0.0074**
F1-Score (↑)	Credit	0.8748 ± 0.0024	0.8767 ± 0.0013	0.8724 ± 0.0014	0.7796 ± 0.0316	0.8751 ± 0.0013	0.8759 ± 0.0020	**0.8801 ± 0.0029**
	Pokec-z	0.7190 ± 0.0016	0.6503 ± 0.0545	**0.7256 ± 0.0080**	0.7102 ± 0.0056	0.7146 ± 0.0030	0.7158 ± 0.0056	0.6723 ± 0.0134
	Pokec-n	0.6761 ± 0.0190	0.6029 ± 0.0663	0.6999 ± 0.0063	**0.7005 ± 0.0091**	0.6924 ± 0.0062	0.6974 ± 0.0083	0.6352 ± 0.0390
	Recidivism	0.9154 ± 0.0042	0.9401 ± 0.0028	0.9167 ± 0.0024	0.7249 ± 0.0128	0.9388 ± 0.0011	0.9320 ± 0.0019	**0.9790 ± 0.0099**
SPD (↓)	Credit	0.0330 ± 0.0131	0.0374 ± 0.0118	0.0430 ± 0.0039	0.0418 ± 0.0237	0.0359 ± 0.0163	0.0334 ± 0.0118	**0.0034 ± 0.0015**
	Pokec-z	0.0658 ± 0.0194	0.0340 ± 0.0109	0.0700 ± 0.0273	**0.0224 ± 0.0190**	0.0587 ± 0.0167	0.0710 ± 0.0185	0.0240 ± 0.0082
	Pokec-n	0.0139 ± 0.0066	0.0170 ± 0.0121	0.0134 ± 0.0066	0.0185 ± 0.0122	0.0181 ± 0.0092	**0.0102 ± 0.0056**	0.0255 ± 0.0142
	Recidivism	0.0746 ± 0.0026	0.0806 ± 0.0021	0.0766 ± 0.0032	0.0565 ± 0.0133	0.0820 ± 0.0017	0.0776 ± 0.0038	**0.0149 ± 0.0099**
EOD (↓)	Credit	0.0261 ± 0.0142	0.0246 ± 0.0113	0.0334 ± 0.0058	0.0435 ± 0.0272	0.0277 ± 0.0142	0.0286 ± 0.0118	**0.0049 ± 0.0019**
	Pokec-z	0.0546 ± 0.0317	0.0469 ± 0.0171	0.0508 ± 0.0332	**0.0147 ± 0.0135**	0.0338 ± 0.0169	0.0500 ± 0.0226	0.0220 ± 0.0114
	Pokec-n	0.0266 ± 0.0153	**0.0059 ± 0.0023**	0.0351 ± 0.0189	0.0365 ± 0.0021	0.0296 ± 0.0076	0.0371 ± 0.0026	0.0530 ± 0.0236
	Recidivism	0.0037 ± 0.0015	0.0142 ± 0.0028	0.0038 ± 0.0004	0.0100 ± 0.0073	0.0138 ± 0.0039	0.0061 ± 0.0046	**0.0034 ± 0.0014**

models regarding fairness on the Pokec-z dataset. For both *Claim 1* and *Claim 2*, we can hence conclude that our results show partial support.

Finally, the authors claim, based on their results, that fairGNN-WOD surpasses the baselines in fairness for all cases and in utility performance for most cases (*Claim 3*). We can indeed say that the utility performance is better for the Credit and Recidivism datasets, with an average increase of 1.94% in accuracy and 2.26% in F1-score on the two datasets against the best baseline. However, on the Pokec datasets, the framework performs significantly worse, with fairGNN-WOD only being able to outperform the GIN baseline. Together with the results discussed regarding fairness, we conclude that this claim is not reproducible, due to the model being outperformed in several areas.

In general, fairGNN-WOD tends to struggle to outperform the baselines in both fairness and utility performance on the Pokec datasets. For the Credit and Recidivism dataset, on the other hand, it shows strong performance in both utility performance and fairness. This indicates that fairGNN-WOD is well-suited for similarity-based graphs, as these are present in these datasets and not in the Pokec datasets.

5.2 Multiple Demographics

The same experiments were run using two and three demographic attributes, and the summarized results against the best possible result of the baselines are documented in Table 2. The full tables can be found in Appendix E. Again, here, we can see similar artifacts as before, where fairGNN-WMD does not outperform the best baseline with regards to utility performance on the Pokec datasets, but does manage to retain competitive performance on the datasets with similarity-based graphs. FairGNN-WMD also shows substantial increases in fairness over all the datasets, with a rare case of being outperformed by one other baseline in the case of two demographic attributes for the Pokec datasets. Especially when incorporating three demographic attributes, fairGNN-WMD significantly outperforms all baselines on fairness, achieving reductions exceeding 65% across all datasets.

This strong performance suggests that fairGNN-WOD generalizes beyond a single sensitive attribute, with fairness gains increasing as the number of demographic attributes grows. Moreover, the best overall performance of fairGNN-WMD is again obtained on the Credit and Recidivism dataset, due to utility performance being retained. This again indicates that the framework is particularly well suited to similarity-based graphs.

Table 2: Percentage change in score of fairGNN-WMD against the best result of the baseline models per dataset and metric. The baseline result serves as the reference value.

Metric	m \ Dataset	Credit	Pokec-z	Pokec-n	Recidivism
Accuracy (↑)	Two	+0.85%	−5.70%	−6.06%	+1.78%
	Three	+1.59%	−1.37%	−3.23%	+1.18%
F1-Score (↑)	Two	+0.13%	−11.89%	−10.92%	+2.54%
	Three	+0.70%	−6.78%	−8.73%	+1.70%
SPD (↓)	Two	−56.57%	−50%	+67.56%	−84.14%
	Three	−89.96%	−86.40%	−76.84%	−83.51%
EOD (↓)	Two	−25.04%	+14.82%	−30.66%	−85.17%
	Three	−74.70%	−83.28%	−75.48%	−66.52%

5.3 Ablation Studies

To validate the assumption made by the authors regarding the independence of $\mathbf{Z}$ from $\mathbf{A}$ given $\mathbf{S}$, and from $\mathbf{X}$ given $\mathbf{S}$ and $\mathbf{A}$, we train another model, where in the first stage we do include $\mathbf{Z}$ as an input for the decoders. This additional connection would be true to the factorization of the joint distribution, without assuming independence. This variant will be denoted by fairGNN-WMD-Z. Based on the comparison between the original fairGNN-WMD and fairGNN-WMD-Z, the validity of this assumption is tested. Moreover, to verify whether the first stage is actually contributing to the performance of fairGNN-WMD, we train the second stage another time using randomly generated demographics. The model resulting from this will be denoted as fairGNN-WMD-R. We conduct the ablation studies for one, two and three missing demographics. Note that for one missing demographics, the ablation studies are directly tested for the original fairGNN-WOD. The summarized results are shown in Figure 3, while the full results are reported in Appendix F.

It can be seen that on utility performance, fairGNN-WMD-R performs better. However, this is an obvious consequence of the framework. The second stage tries to mask any information in the embeddings related to the demographics, and as this information can sometimes be vital for performance, a decrease in accuracy follows. For randomly generated numbers, the second stage would just be trying to mask noise. Therefore, retaining more information, and performing better on accuracy and F1-score. Comparing the fairness metrics on the other hand, it can be seen that the original fairGNN-WMD overall mitigates unfairness more effectively, with only performing worse on EOD with two missing demographics. This comparison points toward the first stage actually contributing to the framework and helping to improve fairness.

Comparing fairGNN-WMD-Z and fairGNN-WMD, we can see that the results are very close to each other. FairGNN-WMD-Z, for lower number of demographics, leads in general to better utility performance and a slight increase in fairness. However, for three missing demographic attributes, fairGNN-WMD performs better on both utility performance and fairness. Hence, including a connection from $\mathbf{Z}$ to the decoders does not lead to a strictly better performance of the model, and if it does, the difference in performance is negligible. As this connection also requires additional trainable parameters, the assumption made by the authors is acceptable.

6 Conclusion

In this paper, we reproduced fairGNN-WOD framework as described by Wang et al. (2025). Then, we tested the model against the same baselines on the same datasets used in the original paper with the addition of the Recidivism dataset, with the aim of verifying the claims made by the authors. We also added new insights to their framework through ablation studies and provide an extension to fairGNN-WOD that is able to account for multiple demographic attributes, called fairGNN-WMD.

We have found that our results show partial evidence for the first two empirical claims made by Wang et al. (2025). Namely, fairGNN-WOD shows improvement in fairness over vanilla graph methods and over other

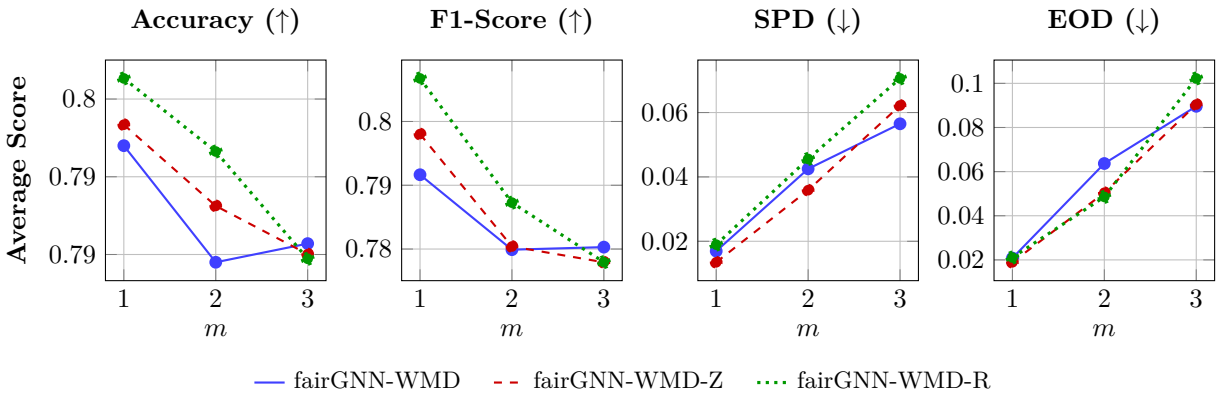

Figure 3: Average scores for utility performance and fairness across datasets for different number of sensitive attributes (m). Note that, fairGNN-WMD with $m = 1$ is the original fairGNN-WOD.

methods adapted from non-graph domains. They also claim that their model in general performs better on utility performance and fairness over all baseline models. However, our results do not show enough evidence for this. FairGNN-WMD shows more support for their claims in a multi-demographics setting, but still does not outperform all the baselines on each dataset.

Furthermore, fairGNN-WOD and its extension to multiple sensitive attributes are well-suited for similarity-based graphs. On both the Credit and Recidivism datasets, the proposed methods outperform all baselines in terms of both utility performance and fairness. Since similarity-based graphs are constructed from tabular data using feature similarity, their neighborhoods are homophilic by design. Strong performance under such conditions indicates that the proposed methods are especially applicable to domains such as fraud detection (Wang et al., 2021) and recommender systems (Gao et al., 2023), where similarity-based graph representations are commonly used.

Our ablation studies, applied to fairGNN-WOD and fairGNN-WMD, show support for the first stage of the framework. At the same time, the model achieves strong performance even when demographic attributes are randomly predicted. This indicates that most of the performance gains stem from the second stage of the framework. Therefore, future work should focus more closely on improving the first stage, to determine whether enhancing this component can further boost overall performance.

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.

Junyi Chai, Taeuk Jang, and Xiaoqian Wang. Fairness without demographics through knowledge distillation. *Advances in Neural Information Processing Systems*, 35:19152–19164, 2022.

April Chen, Ryan A Rossi, Namyong Park, Puja Trivedi, Yu Wang, Tong Yu, Sungchul Kim, Franck Dernoncourt, and Nesreen K Ahmed. Fairness-aware graph neural networks: A survey. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–23, 2024.

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 (WSDM)*, pp. 680–688. ACM, 2021.

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

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, pp. 214–226, 2012.

Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. A survey of graph neural networks for recommender systems: Challenges, methods, and directions. *ACM Transactions on Recommender Systems*, 1(1):1–51, 2023.

Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *Mathematics of Operations Research*, 44(2):377–399, 2019.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Kareem L Jordan and Tina L Freiburger. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. *Journal of Ethnicity in Criminal Justice*, 13(3):179–196, 2015.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

TN Kipf. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

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

Joshua Ka-Wing Lee. *Maximal correlation feature selection and suppression with applications*. PhD thesis, Massachusetts Institute of Technology, 2021.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *International conference on machine learning*, pp. 4212–4221. PMLR, 2019.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016. URL `http://arxiv.org/abs/1611.00712`.

Hongliang Ni, Lei Han, Tong Chen, Shazia Sadiq, and Gianluca Demartini. Fairness without sensitive attributes via knowledge sharing. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1897–1906, 2024.

Kartik Sharma, Yeon-Chang Lee, Sivagami Nambi, Aditya Salian, Shlok Shah, Sang-Wook Kim, and Srijan Kumar. A survey of graph neural networks for social recommender systems. *ACM Computing Surveys*, 56(10):1–34, 2024.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

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

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.

Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications. *arXiv preprint arXiv:2111.15367*, 2021.

Jianian Wang, Sheng Zhang, Yanghua Xiao, and Rui Song. A review on graph neural network methods in financial applications, 2022. URL `https://arxiv.org/abs/2111.15367`.

Zichong Wang, Fang Liu, Shimei Pan, Jun Liu, Fahad Saeed, Meikang Qiu, and Wenbin Zhang. fairgnn-wod: Fair graph learning without demographics. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 556–564, 2025.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Shen Yan, Hsien-te Kao, and Emilio Ferrara. Fair class balancing: Enhancing model fairness without observing sensitive attributes. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1715–1724, 2020.

I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2, Part 1):2473–2480, 2009. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2007.12.020. URL `https://www.sciencedirect.com/science/article/pii/S0957417407006719`.

Tianxiang Zhao, Enyan Dai, Kai Shu, and Suhang Wang. Towards fair classifiers without sensitive attributes: Exploring biases in related features. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 1433–1442, 2022.

Yuchang Zhu, Jintang Li, Zibin Zheng, and Liang Chen. Fair graph representation learning via sensitive attribute disentanglement. In *Proceedings of the ACM Web Conference 2024*, pp. 1182–1192, 2024.

A Derivation of ELBO for the First Stage

A.1 For a Single Missing Demographic Attribute

Observe that using the log-likelihood for $(\mathbf{X}, \mathbf{A})$, we get

$$
\begin{aligned}
\log P(\mathbf{X}, \mathbf{A}) &= \log \iint P(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{Z})\ d\mathbf{S}\ d\mathbf{Z} && \text{(marginalize)} \\
&= \log \iint q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z}) \frac{P(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}\ d\mathbf{S}\ d\mathbf{Z} && \left(\text{multiply by: } \frac{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}\right) \\
&= \log \int q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \left[\int q_\psi(\mathbf{S}|\mathbf{Z}) \frac{P(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})} d\mathbf{S}\right] d\mathbf{Z} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\frac{P(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}\right]\right] \\
&\geq \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log \frac{P(\mathbf{X}, \mathbf{A}, \mathbf{S}, \mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}\right]\right] && \text{(apply Jensen's inequality)} \\
&= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log \frac{P(\mathbf{Z})\ P(\mathbf{S}|\mathbf{Z})\ P(\mathbf{A}|\mathbf{S})\ P(\mathbf{X}|\mathbf{A}, \mathbf{S})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\ q_\psi(\mathbf{S}|\mathbf{Z})}\right]\right] && \text{(factorize joint)} \\
&= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{A}|\mathbf{S}) + \log P(\mathbf{X}|\mathbf{A}, \mathbf{S})\right]\right] && (13) \\
&\quad + \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{Z}) - \log q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\right]\right] && (14) \\
&\quad + \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{S}|\mathbf{Z}) - \log q_\psi(\mathbf{S}|\mathbf{Z})\right]\right]. && (15)
\end{aligned}
$$

Taking a closer look at equation 14, it can be seen that the term within the inner expectation is independent of $\mathbf{S}$, therefore

$$
\begin{aligned}
\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{Z}) - \log q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\right]\right] &= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\log P(\mathbf{Z}) - \log q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\log \frac{P(\mathbf{Z})}{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})}\right] \\
&= -D_{KL}(q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \mid\mid P(\mathbf{Z})). && (16)
\end{aligned}
$$

For equation 15, we can equivalently write

$$
\begin{aligned}
\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{S}|\mathbf{Z}) - \log q_\psi(\mathbf{S}|\mathbf{Z})\right]\right] &= \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log \frac{P(\mathbf{S}|\mathbf{Z})}{q_\psi(\mathbf{S}|\mathbf{Z})}\right]\right] \\
&= -\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[D_{KL}(q_\psi(\mathbf{S}|\mathbf{Z}) \mid\mid P(\mathbf{S}|\mathbf{Z})\right]. && (17)
\end{aligned}
$$

Together equations 20, 14, and 15 form the ELBO used for optimization of stage 1:

$$
\begin{aligned}
\text{ELBO} = &\ \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[\mathbb{E}_{q_\psi(\mathbf{S}|\mathbf{Z})} \left[\log P(\mathbf{A}|\mathbf{S}) + \log P(\mathbf{X}|\mathbf{A}, \mathbf{S})\right]\right] \\
&- D_{KL}(q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \mid\mid P(\mathbf{Z})) \\
&- \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A})} \left[D_{KL}(q_\psi(\mathbf{S}|\mathbf{Z}) \mid\mid P(\mathbf{S}|\mathbf{Z})\right]. && (18)
\end{aligned}
$$

Note that the KL divergence between two Bernoulli distributions can be calculated as (Garivier et al., 2019)

$$
D_{KL}\left(\text{Bernoulli}(p) \mid\mid \text{Bernoulli}(q)\right) = p \cdot \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q} \tag{19}
$$

A.2 For Multiple Missing Demographic Attributes

Using similar steps as above for a single missing demographic attribute, but considering the encoder $q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})$ instead of the original $q_\psi(\mathbf{S}|\mathbf{Z})$, and the joint factorization as in Equation 11, one can obtain:

$$\begin{aligned}\text{ELBO} = &\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[\mathbb{E}_{q_\psi(\mathbf{S}_1,\mathbf{S}_2,...,\mathbf{S}_m|\mathbf{Z})}\left[\log P(\mathbf{A}|\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m) + \log P(\mathbf{X}|\mathbf{A}, \mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m)\right]\right] \\ &- D_{KL}(q_\phi(\mathbf{Z}|\mathbf{X}, \mathbf{A}) \mid\mid P(\mathbf{Z})) \\ &- \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}(q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z}) \mid\mid P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})\right]. \end{aligned} \tag{20}$$

Note that the third term here, the KL divergence between the encoder of the demographic attributes and the conditional prior, stems from the expression

$$\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[\mathbb{E}_{q_\psi(\mathbf{S}_1,\mathbf{S}_2,...,\mathbf{S}_m|\mathbf{Z})}\left[\log P(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z}) - \log q_\psi(\mathbf{S}_1, \mathbf{S}_2, ..., \mathbf{S}_m|\mathbf{Z})\right]\right]. \tag{21}$$

With the assumption made in the introduction of fairGNN-WMD, namely that dempgraphic attributes are independent of each other given $\mathbf{Z}$, this term can equivalently be written as

$$\begin{aligned}&\mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[\mathbb{E}_{q_\psi(\mathbf{S}_1|\mathbf{Z})}\left[\log P(\mathbf{S}_1|\mathbf{Z}) - \log q_\psi(\mathbf{S}_1|\mathbf{Z})\right]\right] + \ldots \\ &+ \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[\mathbb{E}_{q_\psi(\mathbf{S}_m|\mathbf{Z})}\left[\log P(\mathbf{S}_m|\mathbf{Z}) - \log q_\psi(\mathbf{S}_m|\mathbf{Z})\right]\right]\end{aligned} \tag{22}$$

Here each term $q_\psi(\mathbf{S}_i|\mathbf{Z})$ and $P(\mathbf{S}_i|\mathbf{Z})$ are the individual probability outputs for the i'th demographic attribute from the encoders. Now, similarly as before, we can rewrite this as

$$\begin{aligned}&- \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}(q_\psi(\mathbf{S}_1|\mathbf{Z}) \mid\mid P(\mathbf{S}_1|\mathbf{Z})\right] - \ldots - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}(q_\psi(\mathbf{S}_m|\mathbf{Z}) \mid\mid P(\mathbf{S}_m|\mathbf{Z})\right] = \\ &- \sum_{i=1}^{m} \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{A})}\left[D_{KL}(q_\psi(\mathbf{S}_i|\mathbf{Z}) \mid\mid P(\mathbf{S}_i|\mathbf{Z})\right].\end{aligned} \tag{23}$$

Meaning that we can calculate the KL divergence as before in the case of a single demographic attribute, and sum over all m demographic attributes.

B Definition: HGR Maximal Correlation

Definition 1 (HGR Maximal Correlation)**.** *Let $\boldsymbol{X} \in \mathcal{X}$ and $\boldsymbol{Y} \in \mathcal{Y}$ be jointly distributed random variables. Let $f : \mathcal{X} \to \mathbb{R}$ and $g : \mathcal{Y} \to \mathbb{R}$ be the associated maximal correlation functions. Moreover, assume normalization is applied before, such that $\mathbb{E}[f(\boldsymbol{X})] = \mathbb{E}[g(\boldsymbol{Y})] = 0$ and $\mathbb{E}[f^2(\boldsymbol{X})] = \mathbb{E}[g^2(\boldsymbol{Y})] = 1$, then* HGR *maximal correlation is calculated as*

$$\text{HGR}(\boldsymbol{X}, \boldsymbol{Y}) = \sup_{f,g} \mathbb{E}[f(\boldsymbol{X})g(\boldsymbol{Y})]. \tag{24}$$

C Lemma: Full Disentanglement of Channels

Lemma 1. *Assume full disentanglement of channels, so that each channel is independent of the others. Then, it follows that at most one channel representation is related to the demographic attribute.*

Proof. Through proof of contradiction, assume the existence of multiple channel representations related to the demographic attribute. This implies correlations between those channels, which contradicts the initial assumption of full disentanglement. Thus, it can be concluded that at most one channel can be related to the demographic attribute. □

Table 3: Summary statistics on used datasets.

	Credit	Pokec-z	Pokec-n	Recidivism
Total number of nodes	30,000	67,796	66,569	18,876
Number of valid nodes	30,000	10,262 (15%)	8,797 (13%)	18,876
Target label distribution	1: 78% / 0: 22%	1: 54% / 0: 46%	1: 49% / 0: 51%	1: 37.6% / 0: 62.4%
Demographic label distribution	1: 9% / 0: 91%	1: 32% / 0: 68%	1: 32% / 0: 68%	1: 50.6% / 0: 49.4%

D Implementation Details

D.1 Additional Information on Datasets

For credit, the original dataset Yeh & Lien (2009) does not contain a graph structure between users. Hence, users are connected based on similarities on purchase and payment patterns (Agarwal et al., 2021). The age of individuals is binarized into the old and young category, and serves as the demographic attribute.

In the Pokec datasets Takac & Zabovsky (2012), a lot of users have a missing target label, and hence those nodes are only used for message passing. The demographics attribute in both datasets is the region that users are from, with two possible regions.

The Recidivism dataset Jordan & Freiburger (2015) also does not originally contain a graph structure, but connections are made based on the similarity of past criminal records and demographics (Agarwal et al., 2021). The demographic attribute considered is the race.

For all the datasets, continuous features are z-score normalized and categorical variables are normalized to $\{-1, 1\}$. Normalization statistics are computed on the training set only. Moreover, binary features are mapped to $\{-1, 1\}$ accordingly. For all datasets we use a 70%/10%/20% train/val/test split. Table 3 contains further relevant statistics about the data. To extend the models to multiple demographics, for each dataset an additional demographic attribute is chosen. The chosen attributes are whether an individual is single for the Credit dataset, and gender for the Pokec-n, Pokec-z and Recidivism datasets.

D.2 Fairness Metrics

For a single (binary) demographic attribute value s, Statistical Parity Differences (SPD) is defined as

$$\text{SPD} = \left| P(\hat{Y} = 1|s = 1) - P(\hat{Y} = 1|s = 0) \right|, \tag{25}$$

where $\hat{Y}$ denotes the prediction. Equal Opportunity Differences (EOD) is defined as

$$\text{EOD} = \left| P(\hat{Y} = 1|Y = 1, s = 1) - P(\hat{Y} = 1|Y = 1, s = 0) \right|, \tag{26}$$

where Y denotes the ground-truth.

For multiple binary demographic attribute values, it is possible to split individuals in dyadic groups. Each group contains individuals with a certain combination of values for their demographic attributes. Among these groups, it is possible to calculate the same metrics, where SPD is defined as

$$\text{SPD} = \max_d P(\hat{Y} = 1|D = d) - \min_d P(\hat{Y} = 1|D = d), \tag{27}$$

where D denotes the dyadic group. EOD is defined as

$$\text{EOD} = \max_{y \in \{0,1\}} \left(\max_d P(\hat{Y} = 1|Y = y, D = d) - \min_d P(\hat{Y} = 1|Y = y, D = d) \right). \tag{28}$$

D.3 Computational Requirements

The code for the implementation of fairGNN-WOD and fairGNN-WMD, together with the implementation of the baseline models and the datasets, is available on GitHub. The experiments produced an estimated

carbon footprint of 4.15 kg CO_2. This estimate is based on 30 hours of computation on an RTX 3080 GPU using private infrastructure with an assumed carbon efficiency of 0.432 kg/kWh, and is calculated using ML CO_2 Impact.

D.4 Hyperparameters

The fairGNN-WOD presents a set of tunable hyperparameters. In Stage 1, the hyperparameter λ controls the relative importance of the HGR regularization term in the loss function (Equation 3). In Stage 2, the hyperparameters α and β weight the different loss components shown in Equation 10, regulating the trade-off between prediction performance, demographic invariance, and disentanglement. Moreover, the choice of number of channels strikes the balance between utility performance and fairness of the model. In a similar framework, Zhu et al. (2024) found that four channels work the best, and hence this number is also chosen for our framework. Finally, the MMD loss (Equation 5) introduces two additional hyperparameters; the choice of kernel function and the size of the subsample used to estimate the distribution discrepancy. For the fairness-aware baselines the originally proposed hyperparameters by the respective authors were used.

An overview of all the hyperparameters used in our study can be found below.

```
seeds:  14, 17, 49
```

D.4.1 Stage 1

```
hidden_dim:  64
z_dim:  32
s_dim:  16
enc_layers:  3
dec_layers:  3
dropout:  0.0
temp:  0.5
num_neg_per_pos:  3
add_self_loops:  True
normalize:  True
epochs:  150
optimizer_vae:  adam
optimizer_hgr:  adam
lr_vae:  0.001
lr_hgr:  0.001
lambda_hgr:  10.0
hgr_max_steps:  5
detach_hgr_in_min:  False
lambda_adj:  1.0
lambda_x:  1.0
lambda_kl_z:  1.0
lambda_kl_s:  1.0
```

D.4.2 Stage 2

```
hidden_dim:  64
num_channels:  4
num_layers:  2
dropout:  0.5
add_self_loops:  True
normalize:  True
assigner_hidden:  64
epochs:  200
optimizer:  adam
```

```
lr:  0.001
weight_decay:  0.0
clip_grad_norm:  None
alpha:  1.0
beta:  1.0
mmd_sigma:  1.0
mmd_max_samples:  1000
```

D.4.3 Vanilla Baselines

```
epochs:  200
lr:  0.001
weight_decay:  1e-5
model:  gcn #GCN/GIN
num_hidden:  64
num_layers:  2
num_mlp_layers:  2 #Only for GIN
dropout:  0.5
```

E Results of fairGNN-WMD with Multiple Demographic Attributes

Table 4: Results of fairGNN-WMD with baseline methods across used datasets. **Two** demographics are considered for each dataset. For each dataset and each metric, the best result is indicated in bold, whereas the runner-up result is marked with an underline.

Metric	Method / Dataset	GCN	GIN	FairKD	KSMOTE	FairRF	Reckoner	fairGNN-WMD
Accuracy (↑)	Credit	0.7863 ± 0.0034	0.7872 ± 0.0049	0.7854 ± 0.0023	0.7142 ± 0.0158	0.7874 ± 0.0028	0.7883 ± 0.0020	**0.7950 ± 0.0067**
	Pokec-z	0.7133 ± 0.0044	0.6387 ± 0.0456	0.7342 ± 0.007	0.6795 ± 0.0072	0.7308 ± 0.0081	**0.7363 ± 0.0076**	0.6943 ± 0.0097
	Pokec-n	0.7183 ± 0.0075	0.6814 ± 0.0093	0.7312 ± 0.0054	0.7378 ± 0.0098	0.7297 ± 0.0093	**0.7407 ± 0.0042**	0.6958 ± 0.0075
	Recidivism	0.9404 ± 0.0030	0.9555 ± 0.0010	0.9423 ± 0.0020	0.7482 ± 0.0088	0.9558 ± 0.0014	0.9497 ± 0.0021	**0.9728 ± 0.0102**
F1-Score (↑)	Credit	0.8754 ± 0.0019	0.8754 ± 0.0031	0.8740 ± 0.0012	0.8174 ± 0.0137	0.8757 ± 0.0012	0.8769 ± 0.0011	**0.8780 ± 0.0002**
	Pokec-z	0.7048 ± 0.0006	0.6484 ± 0.0624	0.7261 ± 0.0093	0.7063 ± 0.0127	0.7281 ± 0.0068	**0.7310 ± 0.0085**	0.6441 ± 0.0271
	Pokec-n	0.6811 ± 0.0180	0.6553 ± 0.0302	0.6901 ± 0.0098	**0.7113 ± 0.0098**	0.6979 ± 0.0093	0.7028 ± 0.0067	0.6336 ± 0.0270
	Recidivism	0.9171 ± 0.0044	0.9393 ± 0.0011	0.9203 ± 0.0026	0.6994 ± 0.0068	0.9400 ± 0.0018	0.9317 ± 0.0030	**0.9639 ± 0.0132**
SPD (↓)	Credit	0.0489 ± 0.0027	0.0655 ± 0.0268	0.0625 ± 0.0248	0.0426 ± 0.0052	0.0629 ± 0.0205	0.0517 ± 0.0216	**0.0185 ± 0.0062**
	Pokec-z	0.1229 ± 0.0029	0.1191 ± 0.0404	0.1060 ± 0.0136	0.1365 ± 0.0422	0.1056 ± 0.0173	0.1014 ± 0.0158	**0.0507 ± 0.0060**
	Pokec-n	0.0662 ± 0.0055	**0.0373 ± 0.0114**	0.0754 ± 0.0193	0.0905 ± 0.0356	0.0713 ± 0.0278	0.0786 ± 0.0347	0.0625 ± 0.0063
	Recidivism	0.2690 ± 0.0417	0.2721 ± 0.0394	0.2753 ± 0.0363	0.2415 ± 0.0041	0.2852 ± 0.0411	0.2831 ± 0.0432	**0.0383 ± 0.0117**
EOD (↓)	Credit	0.0635 ± 0.0122	0.0958 ± 0.0159	0.0902 ± 0.0256	0.1054 ± 0.0425	0.1301 ± 0.0729	0.1005 ± 0.0059	**0.0476 ± 0.0093**
	Pokec-z	0.1204 ± 0.0270	0.1276 ± 0.0272	0.0857 ± 0.0231	0.1243 ± 0.0263	0.0917 ± 0.0188	**0.0739 ± 0.0070**	0.0984 ± 0.0363
	Pokec-n	0.1752 ± 0.0175	0.1311 ± 0.0184	0.1848 ± 0.0289	0.1817 ± 0.0630	0.1359 ± 0.0361	0.1512 ± 0.0497	**0.0909 ± 0.0171**
	Recidivism	0.1332 ± 0.0610	0.1200 ± 0.0677	0.1457 ± 0.0526	0.2228 ± 0.0410	0.1395 ± 0.0527	0.1406 ± 0.0513	**0.0178 ± 0.0032**

Table 5: Results of fairGNN-WMD with baseline methods across used datasets. **Three** demographics are considered for each dataset. For each dataset and each metric, the best result is indicated in bold, whereas the runner-up result is marked with an underline.

Metric	Dataset \ Method	GCN	GIN	FairKD	KSMOTE	FairRF	Reckoner	fairGNN-WMD
Accuracy (↑)	Credit	0.7865 ± 0.0021	0.7864 ± 0.0047	0.7840 ± 0.0020	0.6402 ± 0.0463	0.7855 ± 0.0041	0.7878 ± 0.0025	**0.8003 ± 0.0097**
	Pokec-z	0.7084 ± 0.0112	0.6850 ± 0.0194	0.7074 ± 0.0075	0.6801 ± 0.0278	**0.7095 ± 0.0034**	0.7087 ± 0.0070	0.6998 ± 0.0042
	Pokec-n	0.7178 ± 0.0080	0.6992 ± 0.0039	**0.7183 ± 0.0088**	0.6720 ± 0.0348	0.7149 ± 0.0042	0.7174 ± 0.0021	0.6951 ± 0.0131
	Recidivism	0.9406 ± 0.0012	0.9566 ± 0.0006	0.9415 ± 0.0010	0.7819 ± 0.0100	0.9558 ± 0.0021	0.9495 ± 0.0015	**0.9679 ± 0.0054**
F1-Score (↑)	Credit	0.8753 ± 0.0009	0.8755 ± 0.0030	0.8732 ± 0.0017	0.7497 ± 0.0448	0.8747 ± 0.0021	0.8760 ± 0.0013	**0.8821 ± 0.0032**
	Pokec-z	0.6976 ± 0.0236	0.6825 ± 0.0146	0.7078 ± 0.0172	0.7050 ± 0.0145	**0.7120 ± 0.0065**	0.7018 ± 0.0094	0.6637 ± 0.0103
	Pokec-n	0.6744 ± 0.0107	0.6519 ± 0.0097	0.6771 ± 0.0065	0.6622 ± 0.0332	0.6776 ± 0.0071	**0.6778 ± 0.0071**	0.6186 ± 0.0453
	Recidivism	0.9173 ± 0.0017	0.9408 ± 0.0007	0.9189 ± 0.0012	0.7338 ± 0.0185	0.9401 ± 0.0028	0.9315 ± 0.0021	**0.9568 ± 0.0070**
SPD (↓)	Credit	0.2663 ± 0.0198	0.2661 ± 0.0417	0.2995 ± 0.0202	0.2514 ± 0.0230	0.2321 ± 0.0344	0.2396 ± 0.0192	**0.0233 ± 0.0059**
	Pokec-z	0.4793 ± 0.0666	0.4619 ± 0.0451	0.4434 ± 0.0414	0.3891 ± 0.0389	0.4564 ± 0.0424	0.4648 ± 0.0647	**0.0529 ± 0.0110**
	Pokec-n	0.4471 ± 0.0108	0.4542 ± 0.0579	0.4503 ± 0.0129	0.3744 ± 0.0927	0.4490 ± 0.0402	0.4621 ± 0.0086	**0.0867 ± 0.0205**
	Recidivism	0.4189 ± 0.0367	0.4139 ± 0.0316	0.4187 ± 0.0399	0.3827 ± 0.0470	0.4312 ± 0.0413	0.4383 ± 0.0516	**0.0631 ± 0.0142**
EOD (↓)	Credit	0.4203 ± 0.0183	0.3159 ± 0.0307	0.4204 ± 0.0183	0.3841 ± 0.0244	0.3075 ± 0.0433	0.3361 ± 0.0797	**0.0778 ± 0.0254**
	Pokec-z	0.6328 ± 0.2013	0.7973 ± 0.2867	0.5885 ± 0.2126	0.7130 ± 0.0307	0.6333 ± 0.1089	0.6769 ± 0.1318	**0.0984 ± 0.0137**
	Pokec-n	0.5694 ± 0.1743	0.6639 ± 0.1662	0.6346 ± 0.1983	0.5286 ± 0.1054	0.6473 ± 0.1349	0.6478 ± 0.1761	**0.1296 ± 0.0163**
	Recidivism	0.4458 ± 0.3352	0.1571 ± 0.0909	0.4489 ± 0.3459	0.5381 ± 0.1311	0.4502 ± 0.3718	0.4517 ± 0.3568	**0.0526 ± 0.0113**

F Results of Ablation Studies

Table 6: Results of fairGNN-WOD with different configurations on the left side. Results for the same ablation studies on fairGNN-WMD with two demographic attributes are presented on the right side. For each dataset, metric and framework, the best result is indicated in bold, whereas the runner-up result is marked with an underline.

Metric	Dataset \ Method	fairGNN-WOD	fairGNN-WOD-Z	fairGNN-WOD-R	fairGNN-WMD	fairGNN-WMD-Z	fairGNN-WMD-R
Accuracy (↑)	Credit	0.7950 ± 0.0066	0.7975 ± 0.0063	**0.7996 ± 0.0026**	0.7950 ± 0.0067	**0.8006 ± 0.0014**	0.7965 ± 0.0053
	Pokec-z	0.7058 ± 0.0086	0.7016 ± 0.0062	**0.7058 ± 0.0076**	0.6943 ± 0.0097	**0.7050 ± 0.0060**	0.7042 ± 0.0050
	Pokec-n	0.7028 ± 0.0090	0.7058 ± 0.0096	**0.7157 ± 0.0140**	0.6958 ± 0.0075	0.7032 ± 0.0087	**0.7038 ± 0.0088**
	Recidivism	0.9843 ± 0.0074	**0.9885 ± 0.0050**	0.9842 ± 0.0078	0.9728 ± 0.0102	0.9637 ± 0.0036	**0.9820 ± 0.0065**
F1-Score (↑)	Credit	0.8801 ± 0.0029	0.8771 ± 0.0047	**0.8812 ± 0.0004**	0.8780 ± 0.0002	**0.8810 ± 0.0022**	0.8771 ± 0.0055
	Pokec-z	0.6723 ± 0.0134	0.6688 ± 0.0210	**0.6965 ± 0.0131**	0.6441 ± 0.0271	0.6684 ± 0.0054	**0.6719 ± 0.0073**
	Pokec-n	0.6352 ± 0.0390	0.6615 ± 0.0202	**0.6705 ± 0.0248**	**0.6336 ± 0.0270**	0.6213 ± 0.0220	0.6243 ± 0.0312
	Recidivism	0.9790 ± 0.0099	**0.9847 ± 0.0067**	0.9788 ± 0.0106	0.9639 ± 0.0132	0.9509 ± 0.0052	**0.9758 ± 0.0087**
SPD (↓)	Credit	**0.0034 ± 0.0015**	0.0046 ± 0.0032	0.0044 ± 0.0042	0.0185 ± 0.0062	**0.0137 ± 0.0009**	0.0139 ± 0.0063
	Pokec-z	0.0240 ± 0.0082	**0.0055 ± 0.0043**	0.0273 ± 0.0222	0.0507 ± 0.0060	0.0611 ± 0.0170	**0.0491 ± 0.0099**
	Pokec-n	0.0255 ± 0.0142	0.0270 ± 0.0184	**0.0225 ± 0.0163**	0.0625 ± 0.0063	**0.0437 ± 0.0075**	0.0532 ± 0.0194
	Recidivism	**0.0149 ± 0.0099**	0.0168 ± 0.0105	0.0216 ± 0.0148	0.0383 ± 0.0117	**0.0250 ± 0.0112**	0.0661 ± 0.0080
EOD (↓)	Credit	0.0049 ± 0.0019	**0.0020 ± 0.0011**	0.0030 ± 0.0016	0.0476 ± 0.0093	0.0412 ± 0.0204	**0.0410 ± 0.0183**
	Pokec-z	0.0220 ± 0.0114	**0.0090 ± 0.0008**	0.0304 ± 0.0235	0.0984 ± 0.0363	0.0793 ± 0.0120	**0.0684 ± 0.0043**
	Pokec-n	0.0530 ± 0.0236	0.0585 ± 0.0333	**0.0434 ± 0.0318**	0.0909 ± 0.0171	**0.0654 ± 0.0216**	0.0661 ± 0.0080
	Recidivism	**0.0034 ± 0.0014**	0.0062 ± 0.0043	0.0075 ± 0.0058	0.0178 ± 0.0032	**0.0158 ± 0.0014**	0.0187 ± 0.0039

Table 7: Results of fairGNN-WMD, using three demographic attributes, with different configurations. For each dataset and each metric, the best result is indicated in bold, whereas the runner-up result is marked with an underline.

Metric	Method / Dataset	fairGNN-WMD	fairGNN-WMD-Z	fairGNN-WMD-R
Accuracy (↑)	Credit	**0.8003 ± 0.0097**	0.7923 ± 0.0028	0.7936 ± 0.0059
	Pokec-z	**0.6998 ± 0.0042**	0.6965 ± 0.0131	0.6965 ± 0.0131
	Pokec-n	0.6951 ± 0.0131	**0.7045 ± 0.0045**	0.6990 ± 0.0050
	Recidivism	0.9679 ± 0.0054	0.9668 ± 0.0053	**0.9698 ± 0.0070**
F1-Score (↑)	Credit	**0.8821 ± 0.0032**	0.8764 ± 0.0016	0.8767 ± 0.0028
	Pokec-z	**0.6637 ± 0.0103**	0.6513 ± 0.0317	0.6513 ± 0.0317
	Pokec-n	0.6186 ± 0.0453	**0.6290 ± 0.0129**	0.6244 ± 0.0200
	Recidivism	0.9568 ± 0.0070	0.9550 ± 0.0071	**0.9594 ± 0.0094**
SPD (↓)	Credit	**0.0233 ± 0.0059**	0.0299 ± 0.0088	0.0292 ± 0.0026
	Pokec-z	**0.0529 ± 0.0110**	0.0784 ± 0.0157	0.0784 ± 0.0157
	Pokec-n	0.0867 ± 0.0205	**0.0752 ± 0.0136**	0.1008 ± 0.0180
	Recidivism	**0.0631 ± 0.0142**	0.0653 ± 0.0147	0.0740 ± 0.0113
EOD (↓)	Credit	0.0778 ± 0.0254	**0.0772 ± 0.0311**	0.1000 ± 0.0234
	Pokec-z	**0.0984 ± 0.0137**	0.1089 ± 0.0171	0.1089 ± 0.0171
	Pokec-n	0.1296 ± 0.0163	**0.1160 ± 0.0047**	0.1419 ± 0.0333
	Recidivism	**0.0526 ± 0.0113**	0.0595 ± 0.0017	0.0581 ± 0.0199

G Results of original authors

Table 8: Comparison between original results (Wang et al., 2025) and our reproduced results for GCN, GIN, and fairGNN-WOD across all metrics.

Metric	Method / Dataset	GCN	GCN **(Ours)**	GIN	GIN **(Ours)**	fairGNN-WOD	fairGNN-WOD **(Ours)**
Accuracy (↑)	Credit	0.781 ± 0.016	0.7855 ± 0.0038	0.787 ± 0.018	0.7878 ± 0.0009	0.754 ± 0.052	0.7950 ± 0.0066
	Pokec-z	0.699 ± 0.024	0.7129 ± 0.0078	0.691 ± 0.015	0.6646 ± 0.0170	0.703 ± 0.041	0.7058 ± 0.0086
	Pokec-n	0.689 ± 0.015	0.7163 ± 0.0049	0.685 ± 0.018	0.6356 ± 0.0225	0.691 ± 0.024	0.7028 ± 0.0090
F1-Score (↑)	Credit	0.868 ± 0.023	0.8748 ± 0.0024	0.877 ± 0.018	0.8767 ± 0.0013	0.861 ± 0.018	0.8801 ± 0.0029
	Pokec-z	0.622 ± 0.012	0.7190 ± 0.0016	0.613 ± 0.007	0.6503 ± 0.0545	0.621 ± 0.032	0.6723 ± 0.0134
	Pokec-n	0.631 ± 0.022	0.6761 ± 0.0190	0.629 ± 0.008	0.6029 ± 0.0663	0.626 ± 0.029	0.6352 ± 0.0390
SPD (↓)	Credit	0.117 ± 0.013	0.0330 ± 0.0131	0.106 ± 0.011	0.0374 ± 0.0118	0.036 ± 0.015	0.0034 ± 0.0015
	Pokec-z	0.075 ± 0.025	0.0658 ± 0.0194	0.061 ± 0.014	0.0340 ± 0.0109	0.028 ± 0.013	0.0240 ± 0.0082
	Pokec-n	0.084 ± 0.013	0.0139 ± 0.0066	0.078 ± 0.017	0.0170 ± 0.0121	0.028 ± 0.013	0.0255 ± 0.0142
EOD (↓)	Credit	0.096 ± 0.017	0.0261 ± 0.0142	0.088 ± 0.013	0.0246 ± 0.0113	0.027 ± 0.013	0.0049 ± 0.0019
	Pokec-z	0.062 ± 0.013	0.0546 ± 0.0317	0.057 ± 0.007	0.0469 ± 0.0171	0.029 ± 0.015	0.0220 ± 0.0114
	Pokec-n	0.078 ± 0.019	0.0266 ± 0.0153	0.071 ± 0.027	0.0059 ± 0.0023	0.038 ± 0.014	0.0530 ± 0.0236

Table 9: Comparison between original results (Wang et al., 2025) and our reproduced results for FairKD and KSMOTE across all metrics.

Metric	Method / Dataset	FairKD	FairKD **(Ours)**	KSMOTE	KSMOTE **(Ours)**
Accuracy (↑)	Credit	0.711 ± 0.012	0.7823 ± 0.0023	0.736 ± 0.009	0.6704 ± 0.0345
	Pokec-z	0.673 ± 0.021	0.7224 ± 0.0069	0.697 ± 0.024	0.6772 ± 0.0028
	Pokec-n	0.663 ± 0.016	0.7361 ± 0.0058	0.669 ± 0.013	0.7104 ± 0.0080
F1-Score (↑)	Credit	0.796 ± 0.023	0.8724 ± 0.0014	0.817 ± 0.012	0.7796 ± 0.0316
	Pokec-z	0.592 ± 0.013	0.7256 ± 0.0080	0.611 ± 0.018	0.7102 ± 0.0056
	Pokec-n	0.603 ± 0.023	0.6999 ± 0.0063	0.611 ± 0.018	0.7005 ± 0.0091
SPD (↓)	Credit	0.094 ± 0.036	0.0430 ± 0.0039	0.071 ± 0.003	0.0418 ± 0.0237
	Pokec-z	0.045 ± 0.014	0.0700 ± 0.0273	0.037 ± 0.017	0.0224 ± 0.0190
	Pokec-n	0.067 ± 0.015	0.0134 ± 0.0066	0.061 ± 0.005	0.0185 ± 0.0122
EOD (↓)	Credit	0.075 ± 0.042	0.0334 ± 0.0058	0.055 ± 0.013	0.0435 ± 0.0272
	Pokec-z	0.048 ± 0.009	0.0508 ± 0.0332	0.039 ± 0.010	0.0147 ± 0.0135
	Pokec-n	0.064 ± 0.013	0.0351 ± 0.0189	0.066 ± 0.013	0.0365 ± 0.0021

Table 10: Comparison between original results (Wang et al., 2025) and our reproduced results for FairRF and Reckoner across all metrics.

Metric	Method / Dataset	FairRF	FairRF **(Ours)**	Reckoner	Reckoner **(Ours)**
Accuracy (↑)	Credit	0.735 ± 0.007	0.7865 ± 0.0026	0.736 ± 0.021	0.7873 ± 0.0034
	Pokec-z	0.690 ± 0.014	0.7178 ± 0.0002	0.692 ± 0.020	0.7202 ± 0.0058
	Pokec-n	0.673 ± 0.013	0.7197 ± 0.0096	0.675 ± 0.028	0.7314 ± 0.0060
F1-Score (↑)	Credit	0.809 ± 0.022	0.8751 ± 0.0013	0.817 ± 0.015	0.8759 ± 0.0020
	Pokec-z	0.617 ± 0.019	0.7146 ± 0.0030	0.603 ± 0.021	0.7158 ± 0.0056
	Pokec-n	0.616 ± 0.032	0.6924 ± 0.0062	0.619 ± 0.032	0.6974 ± 0.0083
SPD (↓)	Credit	0.067 ± 0.017	0.0359 ± 0.0163	0.068 ± 0.017	0.0334 ± 0.0118
	Pokec-z	0.032 ± 0.012	0.0587 ± 0.0167	0.036 ± 0.018	0.0710 ± 0.0185
	Pokec-n	0.056 ± 0.027	0.0181 ± 0.0092	0.042 ± 0.008	0.0102 ± 0.0056
EOD (↓)	Credit	0.057 ± 0.018	0.0277 ± 0.0142	0.055 ± 0.014	0.0286 ± 0.0118
	Pokec-z	0.034 ± 0.012	0.0338 ± 0.0169	0.033 ± 0.010	0.0500 ± 0.0226
	Pokec-n	0.061 ± 0.016	0.0296 ± 0.0076	0.052 ± 0.011	0.0371 ± 0.0026