FairMILE: Towards an Efficient Framework for Fair Graph Representation Learning

Yuntian He  
The Ohio State University  
Columbus, Ohio, USA  
he.1773@osu.edu

Saket Gurukar∗  
Samsung Research America  
Mountain View, California, USA  
saket.gurukar@gmail.com

Srinivasan Parthasarathy  
The Ohio State University  
Columbus, Ohio, USA  
srini@cse.ohio-state.edu

ABSTRACT
Graph representation learning models have demonstrated great capability in many real-world applications. Nevertheless, prior research indicates that these models can learn biased representations leading to discriminatory outcomes. A few works have been proposed to mitigate the bias in graph representations. However, most existing works require exceptional time and computing resources for training and fine-tuning. To this end, we study the problem of efficient fair graph representation learning and propose a novel framework FairMILE. FairMILE is a multi-level paradigm that can efficiently learn graph representations while enforcing fairness and preserving utility. It can work in conjunction with any unsupervised embedding approach and accommodate various fairness constraints. Extensive experiments across different downstream tasks demonstrate that FairMILE significantly outperforms state-of-the-art baselines in terms of running time while achieving a superior trade-off between fairness and utility.

CCS CONCEPTS
• Computing methodologies → Machine learning.

KEYWORDS
Fairness, Machine Learning, Graph Representation Learning

ACM Reference Format:
Yuntian He, Saket Gurukar, and Srinivasan Parthasarathy. 2023. FairMILE: Towards an Efficient Framework for Fair Graph Representation Learning. In Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO ’23), October 30–November 01, 2023, Boston, MA, USA. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3617694.3623231

1 INTRODUCTION
A critical task in graph learning is to learn the hidden representations of the graph, also known as graph embedding. The goal of graph embedding is to preserve both structure properties and node features in the graph. Such embeddings can be used to characterize individual users (e.g. Amazon and Netflix) and to promote new connections (e.g. LinkedIn). Various methods have been developed for this purpose [16, 18, 38, 40], including those based on graph neural networks (GNNs) [19, 28, 48]. Such models have been effective in many real-world applications, such as crime forecasting [23], fraud detection [46], and recommendation [14, 17].

Given the high-stake decision-making scenarios that such models are typically deployed in, it is critical to ensure that the decisions made by these models are fair. Prior studies [1, 10, 41] reveal that graph representation learning models may inherit the bias from the underpinning graph data. A common source of bias is node features which may contain historical bias in sensitive attributes or other correlated attributes [13]. Another cause of bias is the homophily effect - promoting links that may lead to increased segregation. Such bias can lead to a biased distribution in the embedding space [12] and cause unfair treatment towards particular sensitive attributes such as gender and ethnicity [13]. There is a clear need to alleviate such bias, ideally without impacting the bottom line of model performance.

Recent efforts to address this problem seek to enhance fairness by adapting existing GNN models [1, 9, 10, 24, 33]. However, such adaptations often add to the models’ complexity – on large-scale graphs, these models either cannot finish execution in a reasonable amount of time or often result in an out-of-memory error. These concerns are amplified by recent articles that suggest that the training time for many AI models is simply becoming unsustainable [35, 44] with respect to both compute and emission costs. To tackle this issue, a naive solution is to apply scalability improvement techniques such as the multi-level framework [20, 25, 34]. However, these solutions lack fairness considerations. Figure 1 evaluates these approaches in terms of efficiency, fairness, and utility. The results demonstrate that: (1) Prior fairness-aware models are time-consuming for fair graph representation learning; (2) Scalable approaches like MILE [34] cannot enhance the fairness in embeddings. These observations highlight the challenges of balancing efficiency, fairness, and utility in the problem of fair graph representation learning.

In addition to inefficiency, there are some other challenges with existing work. First, some works adapt existing unsupervised graph embedding approaches for fairness [26, 41], but it is challenging to accommodate all such models. Second, many fair representation learning methods only consider a single, binary sensitive attribute, while real-world graphs usually have multiple multi-class sensitive attributes - limiting their applicability. To address the above-mentioned issues, we present Fair Multi Level Embedding framework (FairMILE). FairMILE is a general framework for fair and efficient graph representation learning. It adopts a multi-level framework used by recent scalable embedding
methods [2, 6, 20, 21, 34]. However, unlike other multi-level frameworks, our framework incorporates fairness as a first-class citizen. Our framework is method agnostic in that it can accommodate any unsupervised graph embedding method treating it as a black box. Moreover, unlike a majority of fair graph representation learning models, FairMILE can learn fair embeddings with respect to multiple multi-class sensitive attributes simultaneously. To summarize, our main contributions are:

- **Novelty**: To the best of our knowledge, this is the first work that seeks to improve the efficiency issue present in fair graph representation learning. To that end, we develop a general-purpose framework called FairMILE.

- **Model-agnostic**: FairMILE can easily accommodate any unsupervised graph embedding methods and improve their fairness while preserving the utility.

- **Efficiency and versatility**: Compared with existing approaches, FairMILE successfully improves the efficiency of fair graph representation learning. In addition, FairMILE can achieve fairness towards multiple and non-binary sensitive attributes, which most prior works fail to consider.

- **Evaluation**: We demonstrate both the efficacy and efficiency of FairMILE across both node classification and link prediction settings. Our results show that FairMILE can improve efficiency by up to two orders of magnitude and fairness by several factors while realizing comparable accuracy to competitive strawman.

2 PRELIMINARIES

2.1 Notations

Let $G = (V, E)$ be an undirected graph, where $V$ is the set of nodes, and $E \subseteq V \times V$ is the set of edges. Let $A$ be the graph adjacency matrix, where $A_{u,v}$ denotes the weight of edge $(u,v)$. $A_{u,v} = 0$ means $u,v$ are not connected. $\delta(u)$ denotes the degree of node $u$. The graph also contains a set of sensitive attributes $F$ (e.g., gender and race). Each attribute may have a binary or multi-class value associated with a particular demographic group.

2.2 Fairness Metrics

In this paper, we focus on group fairness and use two metrics for evaluation. These metrics have been widely adopted in prior works [1, 10, 11, 13, 41]. Without loss of generality, we first introduce them in a binary prediction scenario with a binary sensitive attribute, and then we extend them to a general multi-class case.

**Definition 1.** *Demographic Parity* (also known as *statistical parity*) [13] requires that each demographic group should receive an advantageous outcome (i.e., $\hat{Y} > 0$) at the same rate, which is formulated as $P(\hat{Y} = 1|S = 0) = P(\hat{Y} = 1|S = 1)$, where $\hat{Y}$ is the predicted label and $S \in F$ is a binary sensitive attribute. To quantify how demographic parity is achieved, prior works [1, 10, 31] define $\Delta_{DP, binary}$ as

$$\Delta_{DP, binary} = |P(\hat{Y} = 1|S = 0) - P(\hat{Y} = 1|S = 1)|$$

To extend demographic parity for multi-class sensitive attributes, Rahman et al. [41] measure the variance of positive rates among all groups. Here, we further extend it to a multi-class predicted label scenario by averaging the standard deviations (denoted as $\sigma$) across all advantaged classes. The formulation is given as

$$\Delta_{DP} = \frac{1}{|Y^+|} \sum_{y \in Y^+} \sigma \left( \left\{ P(\hat{Y} = y|S = s) : \forall s \right\} \right)$$

where $Y^+$ denotes a set of advantaged classes.

**Definition 2.** *Equality of Opportunity* requires that each demographic group has an identical probability to receive a specific advantageous outcome for its members with this advantaged ground-truth label. In binary classification tasks with respect to a binary sensitive attribute, existing works [1, 10, 11] compute the difference of true positive rates across two groups to measure the equality of opportunity, which is formulated as

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

where $Y$ is the ground truth label. Similarly, we define a new metric $\Delta_{EO}$ to extend equality of opportunity to a general scenario with multi-class labels and attributes:

$$\Delta_{EO} = \frac{1}{|Y^+|} \sum_{y \in Y^+} \sigma \left( \left\{ P(\hat{Y} = y|Y = y,S = s) : \forall s \right\} \right)$$

Note that the previously adopted metrics for binary predicted labels and binary sensitive attributes are a special case of our proposed measures, i.e., $\Delta_{DP} = \frac{1}{2} \Delta_{DP, binary}$ and $\Delta_{EO} = \frac{1}{2} \Delta_{EO, binary}$.

2.3 Problem Statement

**Problem 1.** Given a graph $G = (V, E)$, the embedding dimensionality $d$, and a set of sensitive attributes $F$, the problem of *Fair Graph Representation Learning* aims to learn a fair embedding model $f : V \rightarrow \mathbb{R}^d$ with less inherent bias towards attributes in $F$ where the present bias is measured with $\Delta_{DP}$ and $\Delta_{EO}$.

3 RELATED WORK

3.1 Fairness in Machine Learning

Since machine learning techniques are deployed to make decisions that have societal or ethical implications [7, 46, 47], serious concerns over their fairness are raised. There have been various definitions of fairness in machine learning. In this paper, we focus on the most popular definition *group fairness* [13], which requires that an algorithm should treat each demographic group equally. The
Table 1: Summary of fair graph representation learning methods.

| Method               | Method-agnostic | Multiple Sensitive Attributes | Non-binary Attributes |
|----------------------|-----------------|-------------------------------|-----------------------|
| FairGNN [10]         | ×               | ×                             | ×                     |
| NIFTY [1]            | ×               | ×                             | ×                     |
| FairAdj [33]         | ×               | ×                             | ✓                     |
| FairWalk [41]        | ×               | ×                             | ✓                     |
| CFGE [5]             | ×               | ✓                             | ✓                     |
| EDITS [11]           | ✓               | ×                             | ✓                     |
| FairMILE (This work) | ✓               | ✓                             | ✓                     |

groups are associated with a single or multiple sensitive attributes, such as gender and race. There are also other definitions of fairness including individual fairness [13] and counterfactual fairness [30].

Unfair outcomes are mostly caused by data bias and algorithmic bias [37]. There exists a wide range of biases in data. For example, features like home address can be associated with specific races and lead to unfair decisions indirectly [12]. The design of machine learning algorithms may also unintentionally amplify the bias in data. To address this concern, several fair machine learning algorithms have been proposed in recent years [3, 4, 15, 42, 49]. A comprehensive survey on fair machine learning is given in [37].

3.2 Fair Graph Representation Learning

In the graph context, models trained to realize representations accounting for the connectivity and topology inherent to the network (e.g., homophily bias) can lead to biased representations. Downstream tasks that operate on such representations can lead to unfair recommendations [45], and even biased and unjust outcomes [36]. However, the incorporation of fairness with graph-based learning is challenging because of the non-i.i.d nature of the data and the homophily effect of graph data [10, 12, 22].

Recently, several methods have been proposed to learn fair graph representations. FairGNN [10] leverages adversarial learning to train fair GNNs for node classification. NIFTY [1] adds a fairness loss to the GNN objective as regularization. FairAdj [33] accommodates the VGAE [27] model for fair link prediction. For task-agnostic embedding, FairWalk [41] learns fair embeddings by adapting an embedding algorithm node2vec [16]. Specifically, it modifies the random walk process and adjusts the probability of selecting nodes in each sensitive group for fairness. To consider multiple sensitive attributes, CFGE [5] employs a set of adversaries with the encoder for compositional fairness constraints. Unlike these approaches, EDITS [11] is a pre-processing solution that reduces the bias in graph structure and node attributes, then trains vanilla GNNs on the debiased graph. However, since most methods are GNN-based, they require exceptional time for training and fine-tuning.

In addition to inefficiency, there are three other major drawbacks of existing works. First, some of them are not method-agnostic which means they require non-trivial modifications to the base model for adaption. Second, most existing works are unable to incorporate fairness constraints towards multiple sensitive attributes. Third, some methods cannot handle non-binary sensitive attributes. Table 1 summarizes these approaches together with our proposed work FairMILE. Compared with these approaches, our work can simultaneously (1) accommodate the base model easily while (2) achieve fairness towards multiple non-binary sensitive attributes. Most importantly, we will demonstrate that (3) FairMILE significantly outperforms these baselines in terms of efficiency.

3.3 Scalable Graph Embedding

Many methods for graph embedding have been proposed in recent years, including NetMF [40], DeepWalk [38], and node2vec [16]. Despite their excellent performance on various machine learning tasks, their lack of scalability prohibits them from processing large datasets. Recent research addressed the scalability issue of graph embedding using different methodologies. Some studies leverage high-performance computing techniques [32, 39, 50]. Another group of studies adopts the multi-level framework for better scalability. This framework is widely used for various graph problems [6, 25, 34] and the essential idea is to solve the problem from a smaller coarsened graph. However, none of these methods are fairness-aware. We are the first study that considers fairness based on this framework.

4 METHODOLOGY

We propose a fairness-aware graph embedding framework FairMILE (shown in Figure 2a), which consists of three modules: graph coarsening, base embedding, and refinement. The idea is to first coarsen the original graph into a smaller one, then learn the embeddings of the coarsened graph using the base model, and eventually refine them into the embeddings of the original graph.

There exist several definitions of fairness [13, 30] - all with the shared principle that all subgroups should receive the positive outcome at the same level in a given measure (e.g., positive rate for $\Delta_{DP}$, true positive rate for $\Delta_{EO}$). Since this paper studies unsupervised representation learning, our key idea is to minimize the variance among the representations of different groups. Intuitively, the downstream models trained with such representations (or embeddings) will lead to a decrease in bias. Next, we introduce the functionality of each module and explain how FairMILE enforces fairness in the graph representations while improving efficiency and versatility.

4.1 Graph Coarsening

We develop a new fairness-aware graph coarsening algorithm (shown in Algorithm 1). Given the initial graph $G_0 = G$, it shrinks the graph size by collapsing a group of nodes into a supernode in the output graph $\mathcal{G}_1$. If two nodes are connected in $G_0$, there exists an edge between their supernodes in $\mathcal{G}_1$. As a result, the numbers of nodes and edges are reduced in $\mathcal{G}_1$. After repeating this process $c$ times, we can finally get the coarsened graph $\mathcal{G}_c$.

The key challenges in graph coarsening are two-fold: retaining the structural information for better utility while incorporating fairness toward the sensitive attributes. For utility, we adopt a high-utility coarsening approach Normalized Heavy-Edge Matching (NHEM) [25], which merges two nodes if their normalized edge weight is maximum. Formally, given a node $u \in \mathcal{V}$, NHEM computes the normalized weight of the edge $(u, v) \in \mathcal{E}$ defined as:

$$w(u, v) = \frac{A_{u,v}}{\sqrt{\delta(u)\delta(v)}}$$  (1)
Intuitively, it encourages nodes with fewer connections to match other nodes and penalizes the hub nodes. Node matching allows a supernode to have structural properties from different nodes. However, due to the homophily in many real-world networks, the bias in graph structure can be reinforced if two nodes from the same group are merged. In light of this, we add a new term in the node matching function to reflect the divergence of the sensitive attribute distributions of two nodes for fairness consideration. In FairMILE, we use a matrix $S \in \mathbb{R}^{N \times M}$ to denote the sensitive attribute values, where each row $s_u$ is the attribute value distribution vector for node $u$, and $M$ is the dimensionality of sensitive attribute values. We use these attribute distribution vectors to quantify the divergence between nodes.

Initially, each node in the original graph $G_0$ has a concatenation of one-hot vectors for the attributes. For example, $F$ has two attributes: ‘gender’ (female, male) and ‘race’ (African, Asian, white). The attributes of a female African $u$ can be modeled as $s_u = [1, 0, 1, 0, 0]$. For a node $v$ in a coarser graph $G_i (i > 0)$, we use $s_v$ to denote the distribution of all nodes in $G_0$ merged into it. For example, $s_v = [1, 1, 1, 1, 0]$ in $G_1$ indicates that the supernode $v$ contains one female and one male on attribute ‘gender’, and one African and one Asian on ‘race’. To measure the difference between the sensitive attribute distributions of two nodes, we define the following function based on Kullback–Leibler divergence [29]:

$$\phi(u, v) = 1 - \left(1 + \sum_{j=1}^{M} \frac{s_{u,j}}{\|s_u\|} \log \left( \frac{s_{u,j}}{s_{v,j}} \right) \right)^{-1}$$

(2)

It essentially maps the divergence of two normalized attribute distributions to $[0, 1]$. The higher the score is, the more different their sensitive attributes are. Finally, given a node $u$ of $G_i$, we formulate the node matching policy as follows:

$$\max_{v : (u,v) \in E_i} \left(1 - \lambda_c\right) w(u,v) + \lambda_c \phi(u,v)$$

(3)

where $\lambda_c$ is the weight of fairness in node matching. The objective here is to find the neighbor of node $u$ that maximizes the edge weight and attribute divergence together. Intuitively, a large value of $\lambda_c$ generates more inter-group matching in graph coarsening. Section 5 will empirically show that Algorithm 1 can improve the fairness in graph representations.

4.2 Base Embedding

Like other multi-level frameworks [25, 34], FairMILE applies the base model on the coarsest graph in an agnostic manner. Since the input is the coarsest graph and the output is its node embeddings, it is straightforward that FairMILE accommodates any unsupervised graph embedding method such as DeepWalk [38] or node2vec [16] with no modification required. This step generates the embeddings $E_c$ on the coarsened graph.

4.3 Refinement

In the last phase of FairMILE, we seek to learn the representations of graph $G_0$ from the embeddings of the coarsest graph $G_c$. Generally, we train a fairness-aware refinement model based on graph convolution networks (GCN) [28] to infer the embeddings $E_{c-1}$ of $G_{c-1}$ from $E_c$. Then we iteratively apply it until we get $E_0$.

4.3.1 Model Architecture. Figure 2b shows the architecture of our refinement model. Without loss of generality, the refinement model has a projection layer followed by $l$ GNN layers where the input and output of layer $i \in [1, l]$ are denoted as $H_{i-1}$ and $H_i$. Given two graphs $G_j$ and $G_{j+1}$, we initialize $H_0$ by projecting the embeddings of supernodes in $G_{j+1}$ to its associated nodes in $G_j$. Note that if two nodes that have different sensitive attribute values are merged...
into $G_{j+1}$, they share the same initial embeddings in $G_j$, which mitigates the potential bias in learned representations.

In each layer, we use a normalized adjacency matrix $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ for message passing. Here we drop the notation referring to a specific graph for clarity. To take the sensitive attributes into consideration, we concatenate the input of each layer with the row-normalized sensitive attribute matrix $\tilde{S}$. Formally, the $i$-th convolution layer in our refinement model can be formulated as

$$H_i = \tanh \left( \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} (H_{i-1} \parallel \tilde{S}) \Theta_i \right)$$  \hspace{1cm} (4)

where $\Theta_i$ is the trainable linear transformation matrix of layer $i$. Finally, we can infer $G_j$’s node representations $E_j = H_j/\|H_j\|_2$.

### 4.3.2 Training objectives

To achieve a trade-off between utility and fairness, we have two distinct objectives for each of them. For utility, we expect the refined representations to be close to the input so that matched nodes are still close to each other in the embedding space after refinement. Therefore, we minimize the difference between the projected embeddings and the predicted ones generated by the refinement model, which is defined as

$$L_u = \frac{1}{|V_c|} \|H_0 - H_t\|^2$$  \hspace{1cm} (5)

$L_u$ is the utility loss in our framework. To improve fairness, we encourage nodes with different sensitive attribute values to be closer in the embedding space. Specifically, we create a subset of edges $E'_c$ that consists of links between two nodes with significantly different sensitive attributes, which can be formulated as

$$E'_c = \{(u, v) : (u, v) \in E_c \land \phi(u, v) \geq \gamma\}$$

where $\gamma$ is a threshold parameter for attribute divergence. Then we use the learned representations to reconstruct an adjacency matrix that represents the edges in the embedding space. To reward the links between diverse attribute groups, we use the Hadamard product of the reconstruction matrix and the adjacency matrix of $E'_c$ (denoted as $\mathcal{A}'$) as our fairness objective. As a result, the trained model generates similar embeddings for inter-group nodes. The fairness loss ($L_f$) is formally defined as

$$L_f = -\frac{1}{|E_c|} \left| \text{sigmoid}(H_tH_t^\top) \odot \mathcal{A}' \right|$$  \hspace{1cm} (6)

The multiplication of the embedding with a sigmoid activation (denoted by $\text{sigmoid}(H_tH_t^\top)$) reconstructs the entire $N \times N$ matrix. Its element-wise product with $\mathcal{A}'$ performs the masking operation to select the node pairs with diverse sensitive attributes. The negative minimization of $L_f$ ensures that the similarity of the selected node pairs – measured by the dot products of their embeddings – is increased. This loss ensures that the embeddings learned by the refinement model of the node pairs with diverse sensitive attributes are proximal to each other.

Combining the utility and fairness loss functions, the overall training objective of our refinement model is

$$\min_{\{\Theta_i, \forall i \in [1, l]\}} (1 - \lambda_r) L_u + \lambda_r L_f$$  \hspace{1cm} (7)

where $\lambda_r \in [0, 1]$ controls the weight of fairness objective.

### 4.4 Theoretical analysis

**Time complexity:** The time complexity of FairMILE depends on the selected embedding approach. Note that such approaches typically have a time complexity of at least $O(d|V_0|)$ [8], for example, the time complexity of DeepWalk is $O(d|V_0| \log |V_0|)$. Considering the number of nodes can be reduced by up to half after each time of coarsening (observed in Section 5.4), the efficiency is significantly improved by embedding the coarsened graph. Apart from embedding the coarsened graph, FairMILE spends additional $O((\ell(d + M)) (|E_0| + d|V_0|))$ time on coarsening and refinement. Given that $d, M \ll |V_0|$, the additional time of these two phases is typically much less than the reduced time of base embedding - empirically verified in Section 5.4. Additional details on complexity analysis are included in our supplement.

**Fairness:** We prove that the difference between the mean representations of different demographic groups is bounded depending on the network topology.

**Theorem 1.** When $L_f$ is minimized, the 2-norm of the difference between the mean embeddings of any two demographic groups regarding a given sensitive attribute is bounded by

$$\|\mu_p - \mu_q\|_2 \leq 2(1 - \min(p_q))$$

where $p, q$ are any two different values of the given sensitive attribute. For $i \in \{p, q\}$, $\mu_i$ denotes the mean embedding values of nodes from group $i$, and $\beta_i$ denotes the ratio of nodes from group $i$ that have at least one inter-group edge.

Theorem 1 shows that the difference between the mean embeddings of two groups depends on the ratio of inter-group connected nodes in each group, which is typically large. For example, among the datasets used in our experiments, the minimum $\beta$ is 0.676 in Credit and 0.958 in German, respectively. When the mean embeddings of different demographic groups are close to each other, they have similar representations and therefore with a high likelihood, they will receive similar outcomes in the downstream task. The full proof of Theorem 1 is provided in our supplement.

### 5 EXPERIMENTS

#### 5.1 Experiment Setup

**Datasets:** We examine the performance of FairMILE on both node classification and link prediction tasks. Our experiments are conducted on seven real-world datasets from different application scenarios widely used in the fairness literature [1, 10, 11, 33, 43]. Statistics of the datasets are shown in Table 2, where $|\mathcal{V}|$ denotes the number of predicted labels, and $|\mathcal{S}|$ denotes the number of sensitive attribute values. For details, please refer to our supplement.

**Metrics:** To quantify the prediction performance in node classification, we use AUROC, F1-score (for binary class problems), and Micro-F1 (for multi-class problems) as our utility metric. To measure the group fairness, we use $\Delta_{DP}$ and $\Delta_{EO}$ described in Section 3 as our fairness metrics. We also report the end-to-end running time in seconds to show the efficiency of all methods.

For link prediction, following prior works [33, 43], we use AUROC, Average Precision (AP), and accuracy as the utility metrics, and compute the disparity in expected prediction scores between intra-group and inter-group links. Specifically, the fairness metrics...
Table 2: Statistics of datasets (NC denotes Node Classification, and LP denotes Link Prediction).

| Dataset | Task   | # Nodes | # Edges | # Features | Label (|\mathcal{Y}|) | Sensitive Attributes (|\mathcal{S}|) |
|---------|--------|---------|---------|------------|----------------------|-------------------------------|
| German  | NC     | 1,000   | 22,242  | 27         | credit risk (2)       | gender (2)                    |
| Recidivism | NC   | 18,876  | 321,308 | 18         | crime (2)             | race (2)                      |
| Credit  | NC     | 30,000  | 1,436,858 | 13      | payment default (2)   | age (2)                       |
| Poken-n | LP     | 2,708   | 5,278   | 1433       | citation (2)          | paper category (7)            |
| Cora    | LP     | 3,312   | 4,660   | 3703       | citation (2)          | paper category (6)            |
| Citeseer | LP    | 19,717  | 44,338  | 500        | citation (2)          | paper category (3)            |

are formulated as:

\[\Delta_{DP, LP} = \mathbb{E}_{(u,v) \sim \mathcal{V} \times \mathcal{V}} \{\hat{Y}|S(u) = S(v)\} - \mathbb{E}_{(u,v) \sim \mathcal{V} \times \mathcal{V}} \{\hat{Y}|S(u) \neq S(v)\}\]

\[\Delta_{EO, LP} = \mathbb{E}_{(u,v) \sim \mathcal{V} \times \mathcal{V}} \{\hat{Y}|(u,v) \in \mathcal{E}, S(u) = S(v)\} - \mathbb{E}_{(u,v) \sim \mathcal{V} \times \mathcal{V}} \{\hat{Y}|(u,v) \in \mathcal{E}, S(u) \neq S(v)\}\]

where \(\hat{Y} \in [0,1]\) is the link prediction score.

**Baselines:** Our baselines include 1) Specialized approaches: For node classification, we use the vanilla GCN [28] and three state-of-the-art fair node classification methods (NIFTY [1], FairGNN [10], and EDITS [11]) with GCN as their base model. For link prediction, we use VGAE [27] and FairAdj [33] with VGAE as the base predictor in our comparative experiments. In addition, we adapt CFGE [5] for both tasks which is the only baseline that accommodates multiple sensitive attributes. 2) Graph embedding approaches: We choose three popular unsupervised graph embedding techniques: NetMF [40], DeepWalk [38], and Node2vec [16]. We also include FairWalk [41], which is essentially a fairness-aware adaption of Node2vec. Note that these approaches can be used in both downstream tasks. 3) Our framework: We let FairMILE run with the three embedding approaches above for evaluation.

**Parameters and environments:** All hyperparameters of specialized approaches are set following the authors’ instructions. In particular, unless otherwise specified, we set \(\lambda = 50\) in CFGE for a better tradeoff between fairness and utility. The dimensionality \(d\) of representations for graph embedding approaches and our work is set to 128. In FairMILE, we set \(\lambda_r = 0.5\) for graph coarsening. For refinement, we train a two-layer model for 200 epochs with \(\lambda_r = 0.5, \gamma = 0.5\), and the learning rate of \(1 \times 10^{-3}\) on all datasets. To evaluate unsupervised approaches for node classification and all approaches for link prediction, we train a linear classifier on the learned embeddings (for node classification) or the Hadamard products of embeddings of sampled node pairs (for link predictions). For the task of link prediction, we randomly sample 10% of edges to build test sets and remove them from the training data, then we add the same number of negative samples as positive edges in the training and test sets, respectively. All methods are evaluated on the same test sets and trained on CPUs for fair comparisons. Experiments are conducted on a Linux machine with a 28-core Intel Xeon E5-2680 CPU and 128GB RAM. We report the average results of 5 runs with different data splits. For reproducibility, our codes and data are available\(^1\).

5.2 Results for Node Classification

We first compare our work with specialized GNN-based approaches and unsupervised graph embedding baselines on three datasets with a single binary sensitive attribute. Specifically, we address the following questions: Q1: Does FairMILE improve the fairness of a base embedding method? How is the fairness of FairMILE compared with specialized approaches? Q2: Does FairMILE outperform other baselines in terms of efficiency? Q3: Does FairMILE retain the embedding’s utility while improving its fairness and efficiency? Table 3 only presents the results on two datasets (German and Credit). The results on Recidivism are included in our supplement. Methods are categorized into different groups by their base models, and the optimal performance in each group is highlighted in bold.

**A1: FairMILE achieves better fairness scores.** Compared with graph embedding approaches without fairness consideration (i.e., NetMF, DeepWalk, and Node2vec), FairMILE on top of them always has lower \(\Delta_{DP}\) and \(\Delta_{EO}\). In German, FairMILE decreases the \(\Delta_{DP}\) scores of Node2vec and DeepWalk by 83.7% and 90.7%, respectively. In terms of \(\Delta_{EO}\), FairMILE improves the fairness of Node2vec and DeepWalk by 84.0% and 96.6%. When choosing NetMF as the base model, FairMILE is optimal (zero) on both fairness metrics, which indicates that the learned representations lead to a perfectly fair classification. Compared with FairWalk (the fair adoption of Node2vec), FairMILE-Node2vec improves \(\Delta_{DP}\) by 49.7% on Credit while FairWalk only improves by 4.8%. Similar results are also observed on other datasets, which reveals that FairMILE successfully mitigates the bias in general-purpose graph embedding baselines.

We also evaluate specialized methods (i.e., FairGNN, NIFTY, EDITS, and CFGE) and observe that FairMILE achieves comparable or better fairness with respect to these approaches. FairGNN outperforms the other specialized methods on all datasets by gaining the largest improvements on both \(\Delta_{DP}\) and \(\Delta_{EO}\) with respect to Vanilla GCN. It means that their methodology of adversarial training is more effective than the regularization adopted by NIFTY, and EDITS suffers from its task agnosticism. On German and Credit, the best fairness metrics of FairMILE are comparable to or better than FairGNN and other specialized techniques. These results demonstrate that FairMILE is effective in reducing the bias in graph representations compared to the state-of-the-art models.

**A2: FairMILE is more efficient than other baselines.** First, FairMILE outperforms the GNN-based specialized approaches on all datasets in terms of efficiency. In German, they take up to 2 minutes for training, while FairMILE can finish in only 6 seconds. The difference becomes more significant on larger datasets. When CFGE needs more than 3 hours on Credit, FairMILE finishes within 2 minutes which is 110.6-137.7\times faster. On the other hand, FairMILE also

\(^1\)https://github.com/heyuntian/FairMILE
improves the efficiency of all base embedding methods. In Credit, FairMILE on top of NetMF saves up to 80% of the running time of vanilla NetMF. The improvement in German is sometimes invisible because German is a small graph. But FairMILE can still finish within seconds. Compared with FairWalk, FairMILE-Node2vec is always faster.

**A3: FairMILE learns quality graph representations.** We observe the quality of learned representations through the AUROC and F1 scores. With respect to the base embedding methods, FairMILE has a similar performance on both utility metrics which is fairly remarkable given that FairMILE significantly improves fairness. An interesting observation is that while Vanilla GCN, graph embedding approaches, and FairMILE have a similar performance in terms of utility, the supervised GCN-based fair approaches have lower AUROC or F1 scores on one or both datasets, which reveals that these approaches require fine-tuning to perform well in terms of utility while enforcing fairness.

In summary, FairMILE in conjunction with popular base embedding approaches can compete or improve on the fairness criteria with various specialized methods while outperforming them significantly in terms of efficiency and retaining comparable utility.

### 5.3 Fairness towards Multiple Sensitive Attributes

To explore how FairMILE learns fair representations towards multiple sensitive attributes, we conduct an experiment on the Pokec-n dataset. Pokec-n has a multi-class predicted label and two sensitive attributes. Most baselines cannot process such datasets since they restrictively only cater to the case of a single binary label or sensitive attribute. CFGE [5] is the only baseline that accommodates multiple sensitive attributes. Therefore we compare FairMILE with CFGE [5] and graph embedding methods, and we set $c = 4$ for our work. Note that FairWalk can consider only one sensitive attribute at a time, thus we run it with each one of the two sensitive attributes and show the results of both runs. In Table 4, we show the results of all methods on Pokec-n in terms of utility, efficiency, and fairness with respect to two sensitive attributes. First of all, with respect to standard embedding methods, FairMILE improves the efficiency and the fairness towards both sensitive attributes while the utility remains competitive. For example, FairMILE reduces the $\Delta_{DP}$ of DeepWalk by 65.8% on ‘region’ and 51.1% on ‘gender’, respectively. Second, although FairWalk also fulfills the fairness towards the assigned attribute, FairMILE-Node2vec has a better fairness score on both attributes. Third, FairMILE significantly outperforms CFGE on efficiency given that they have comparable utility and fairness performance. To study the performance of CFGE, we tune the hyperparameter $\lambda$ which controls the strength of fairness. We pick $\lambda$ from 1 to 100. Going beyond 100 we find the drop in utility exceeds 20% which is often unacceptable. When the constraint is strict ($\lambda = 100$), CFGE has better fairness outcomes at a significant cost to the utility. For $\lambda = 1$, 10 and 50, CFGE and FairMILE have competitive performance in terms of fairness and utility trade-off. However, while CFGE takes around 9500 seconds, FairMILE finishes in only 200-300 seconds, which is up to 46× faster.

**Table 3: Comparison on utility, fairness, and efficiency metrics in node classification between FairMILE and other baselines.**

| Dataset | Method               | AUROC ($\uparrow$) | F1 ($\uparrow$) | $\Delta_{DP}$ ($\downarrow$) | $\Delta_{EO}$ ($\downarrow$) | Time ($\downarrow$) |
|---------|----------------------|---------------------|----------------|-------------------------------|-------------------------------|--------------------|
| German  | NetMF                | 65.16 ± 2.45        | 80.63 ± 1.10   | 5.71 ± 2.89                   | 3.66 ± 2.11                   | 2.48               |
|         | FairMILE-NetMF       | 61.93 ± 3.38        | 82.35 ± 0.00   | 0.00 ± 0.00                   | 0.00 ± 0.00                   | 6.31               |
|         | DeepWalk             | 58.54 ± 4.43        | 75.78 ± 1.49   | 7.22 ± 3.86                   | 7.69 ± 3.26                   | 16.99              |
|         | FairMILE-DeepWalk    | 63.31 ± 3.63        | 82.40 ± 0.33   | 0.67 ± 0.88                   | 0.26 ± 0.39                   | 7.84               |
|         | Node2vec             | 63.37 ± 3.77        | 78.69 ± 1.25   | 3.69 ± 2.60                   | 2.75 ± 1.34                   | 12.76              |
|         | FairWalk             | 63.98 ± 2.07        | 77.64 ± 1.62   | 3.67 ± 2.74                   | 3.28 ± 2.50                   | 11.93              |
|         | FairMILE-Node2vec    | 62.00 ± 2.59        | 82.32 ± 0.20   | 0.60 ± 0.96                   | 0.44 ± 0.41                   | 8.29               |
|         | Vanilla GCN          | 64.73 ± 7.20        | 77.93 ± 3.53   | 16.27 ± 5.86                  | 13.28 ± 5.06                  | 23.75              |
|         | FairGNN              | 53.12 ± 5.73        | 82.35 ± 0.00   | 0.00 ± 0.00                   | 0.00 ± 0.00                   | 136.29             |
|         | NIFTY                | 56.65 ± 6.84        | 81.35 ± 1.54   | 1.20 ± 1.45                   | 0.83 ± 1.20                   | 91.05              |
|         | EDITS                | 64.93 ± 2.90        | 79.64 ± 2.27   | 5.00 ± 3.38                   | 2.76 ± 2.26                   | 84.24              |
|         | CFGE                 | 64.38 ± 0.77        | 81.59 ± 0.33   | 4.54 ± 2.91                   | 4.30 ± 1.84                   | 3900.11            |
| Credit  | NetMF                | 74.93 ± 0.43        | 88.36 ± 0.08   | 2.66 ± 0.55                   | 1.34 ± 0.93                   | 240.38             |
|         | FairMILE-NetMF       | 74.69 ± 0.43        | 88.31 ± 0.08   | 0.68 ± 0.50                   | 0.68 ± 0.66                   | 90.28              |
|         | DeepWalk             | 75.09 ± 0.39        | 88.36 ± 0.15   | 2.50 ± 0.54                   | 1.81 ± 0.71                   | 570.35             |
|         | FairMILE-DeepWalk    | 74.60 ± 0.53        | 88.31 ± 0.11   | 1.34 ± 0.57                   | 1.05 ± 0.70                   | 112.39             |
|         | Node2vec             | 74.95 ± 0.36        | 88.26 ± 0.18   | 2.92 ± 0.41                   | 1.93 ± 0.87                   | 249.92             |
|         | FairWalk             | 74.88 ± 0.49        | 88.25 ± 0.15   | 2.78 ± 0.47                   | 1.58 ± 0.81                   | 261.23             |
|         | FairMILE-Node2vec    | 74.01 ± 0.47        | 87.99 ± 0.16   | 1.47 ± 0.74                   | 1.33 ± 0.64                   | 105.43             |
|         | Vanilla GCN          | 72.83 ± 2.83        | 82.79 ± 1.76   | 5.77 ± 0.32                   | 4.36 ± 0.59                   | 2265.58            |
|         | FairGNN              | 71.14 ± 4.58        | 83.29 ± 3.27   | 1.40 ± 0.97                   | 1.21 ± 0.50                   | 3201.72            |
|         | NIFTY                | 72.43 ± 0.68        | 81.80 ± 0.37   | 5.52 ± 0.33                   | 4.35 ± 0.68                   | 7755.00            |
|         | EDITS                | 73.22 ± 0.77        | 81.55 ± 0.30   | 5.36 ± 0.42                   | 4.17 ± 0.65                   | 9078.12            |
|         | CFGE                 | 70.51 ± 1.31        | 87.85 ± 0.13   | 2.99 ± 0.47                   | 1.47 ± 0.18                   | 12430.37           |
We observe that increasing $c$ demonstrates our effectiveness in fairness improvement. We also baseline of scalable embedding methods without fairness consideration. Additionally, we choose MILE [34] as a refinement, we observe the change in performance when we remove the fairness-aware designs. To evaluate the effectiveness of our fairness-aware modules for graph coarsening and refinement, we observe the change in performance when we remove the fairness-aware designs. For example, vanilla NetMF and CFGE. Finally, FairMILE has a similar or slightly better utility compared with other approaches. This demonstrates that FairMILE consistently achieves the trade-off between fairness, efficiency, and utility in a different downstream task.

## 5.4 Ablation Study

In the ablation study, we showcase the impact of coarsen level $c$ on FairMILE’s performance and the effectiveness of its modules in the fairness of learned representations. In our supplement, we also play with hyperparameters $\lambda_r$ and $\lambda_e$ to study the trade-off between fairness and effectiveness.

### 5.4.1 Impact of coarsening.

We vary the coarsen level $c$ to observe its impact on graph sizes and model performance. Table 5 shows the results with NetMF on the Credit dataset. Other results are similar and can be found in our supplement. Specifically, we study:

**P1: How the graph changes after each time of coarsening.**

We observe that increasing $c$ exponentially reduces the numbers of nodes and edges, which corroborates the analysis in Section 4.4.

**P2: How FairMILE improves the efficiency by coarsening.**

Generally, the efficiency is significantly improved when $c$ increases. A small $c$ (e.g., $c = 1$) may make FairMILE slower because the time of coarsening and refinement outweighs the saved time of learning embeddings when the coarsened graph is not small enough.

**P3: How the fairness evolves with varying $c$.**

In terms of fairness in the downstream task, we observe that increasing $c$ can visibly improve the fairness of representations. For example, vanilla NetMF has $\Delta_{DP} = 2.66$ and $\Delta_{EO} = 1.34$, which is improved to $\Delta_{DP} = 0.68$ and $\Delta_{EO} = 0.68$ by FairMILE ($c = 4$).

**P4: How the utility is impacted by the information loss.**

We find increasing $c$ leads to a slight decrease in AUROC and F1 scores. The AUROC score only decreases by 0.3% after FairMILE coarsens the graph 4 times. In some cases, FairMILE achieves a better utility than the base embedding method (i.e., with $c = 1$). Given the little cost of utility, we suggest using a large $c$ for the sake of fairness and efficiency.

### 5.4.2 Effectiveness of fairness-aware modules.

To evaluate the effectiveness of our fairness-aware modules for graph coarsening and refinement, we observe the change in performance when we remove the fairness-aware designs. Additionally, we choose MILE [34] as a baseline of scalable embedding methods without fairness considerations. Table 6 reports the results on three datasets. The result shows that FairMILE outperforms MILE in terms of fairness, which demonstrates our effectiveness in fairness improvement. We also notice that the fairness scores decline on all datasets when removing either fairness-aware design in FairMILE, indicating that these designs effectively mitigate the bias in the learned embeddings. Furthermore, the little differences between the utility scores of these methods demonstrate that FairMILE is able to improve fairness without impacting utility when compared with MILE. However, this improved fairness does come at some cost to efficiency w.r.t MILE (which is always faster than FairMILE).

## 5.5 Results for Link Prediction

We next evaluate FairMILE in the context of link prediction. Table 7 only shows the results on Cora and Citeseer. Full results and detailed analysis are in our supplement. First, FairMILE effectively mitigates the bias in link prediction. Our framework has lower $\Delta_{DP, LP}$ and $\Delta_{EO, LP}$ than the specialized approaches and graph embedding approaches. On the other hand, FairMILE is more efficient than other baselines, especially the fairness-aware competitors FairAdj and CFGE. Finally, FairMILE has a similar or slightly better utility compared with other approaches. This demonstrates that FairMILE consistently achieves the trade-off between fairness, efficiency, and utility in a different downstream task.

## 6 CONCLUSION

In this paper, we study the problem of fair graph representation learning and propose a general framework FairMILE, which can incorporate fairness considerations with any unsupervised graph embedding algorithms and learn fair embeddings towards one or multiple sensitive attributes. We conduct comprehensive experiments to demonstrate that with respect to state-of-the-art techniques for fair graph representation learning, our work achieves similar or better performance in terms of utility and fairness, while FairMILE can significantly outperform them on the axis of efficiency (up to two orders of magnitude faster). Planned future work includes evaluating the use of FairMILE for real-world graph-based model auditing [36] in deployed online settings.

## ACKNOWLEDGMENTS

This material is supported by the National Science Foundation (NSF) under grants OAC-2018627, CCF-2028944, and CNS-2112471. Any
Table 5: Impact of coarsen level on graph sizes and FairMILE’s performance.

| Metric          | Vanilla NetMF | c = 1          | c = 2          | c = 3          | c = 4          |
|-----------------|---------------|----------------|----------------|----------------|----------------|
| # Nodes         | 30000         | 15033          | 7544           | 3789           | 1899           |
| # Edges         | 1.44M         | 550K           | 256K           | 138K           | 83K            |
| Time (s)        | 240.38        | 487.60         | 208.75         | 139.79         | 90.28          |
| $\Delta DP (\%)$| 2.66 ± 0.55   | 2.23 ± 0.58    | 2.07 ± 0.22    | 1.88 ± 0.37    | 0.68 ± 0.30    |
| $\Delta EO (\%)$| 1.34 ± 0.93   | 1.22 ± 0.80    | 1.14 ± 0.76    | 1.09 ± 0.69    | 0.68 ± 0.66    |
| AUROC (F1)      | 74.93 ± 0.43  | 75.12 ± 0.52   | 74.95 ± 0.40   | 74.80 ± 0.41   | 74.69 ± 0.43   |

Table 6: Ablation study of each module’s effectiveness in fairness (with NetMF).

| Dataset | Method                                      | AUROC (F1) | F1 (F1) | $\Delta DP (\%)$ | $\Delta EO (\%)$ | Time (s) |
|---------|---------------------------------------------|------------|----------|-------------------|-------------------|-----------|
| German  | FairMILE                                    | 61.93 ± 3.38| 82.35 ± 0.00| 0.00 ± 0.00       | 0.00 ± 0.00       | 6.31      |
|         | FairMILE w/o fair coarsening                | 63.61 ± 2.91| 82.27 ± 0.58| 2.20 ± 0.98       | 1.47 ± 1.54       | 5.98      |
|         | FairMILE w/o fair refinement                | 63.37 ± 2.91| 81.08 ± 0.60| 2.94 ± 1.58       | 2.02 ± 1.06       | 5.64      |
|         | MILE                                         | 63.02 ± 2.76| 80.74 ± 0.76| 3.28 ± 2.10       | 2.01 ± 0.97       | 5.06      |
| Recidivism | FairMILE                                    | 89.52 ± 0.50| 77.65 ± 0.47| 2.81 ± 0.50       | 0.75 ± 0.55       | 29.66     |
|         | FairMILE w/o fair coarsening                | 90.23 ± 0.43| 79.62 ± 0.64| 3.10 ± 0.33       | 1.34 ± 0.61       | 22.36     |
|         | FairMILE w/o fair refinement                | 90.25 ± 0.23| 79.94 ± 0.50| 3.27 ± 0.36       | 1.30 ± 0.66       | 26.01     |
|         | MILE                                         | 90.75 ± 0.30| 80.46 ± 0.64| 3.32 ± 0.32       | 1.43 ± 0.65       | 18.42     |
| Credit  | FairMILE                                    | 74.69 ± 0.43| 88.31 ± 0.08| 0.68 ± 0.50       | 0.68 ± 0.66       | 90.28     |
|         | FairMILE w/o fair coarsening                | 74.65 ± 0.34| 88.30 ± 0.12| 1.50 ± 0.52       | 0.92 ± 0.74       | 57.21     |
|         | FairMILE w/o fair refinement                | 74.42 ± 0.33| 88.25 ± 0.15| 3.15 ± 0.49       | 1.71 ± 0.68       | 82.01     |
|         | MILE                                         | 74.65 ± 0.41| 88.33 ± 0.12| 2.54 ± 0.48       | 1.38 ± 0.81       | 49.98     |

Table 7: Comparison on utility, fairness, and efficiency metrics in link prediction between FairMILE and other baselines.

| Dataset | Method             | AUROC (F1) | AP (F1) | Accuracy (F1) | $\Delta DP, LP (\%)$ | $\Delta EO, LP (\%)$ | Time (s) |
|---------|--------------------|------------|---------|---------------|----------------------|----------------------|-----------|
| Cora    | VGAE               | 90.86 ± 0.79| 92.81 ± 0.89| 82.68 ± 1.19  | 47.51 ± 2.47         | 24.12 ± 3.29         | 15.17     |
|         | FairAdjT2=2        | 89.56 ± 1.06| 91.31 ± 1.21| 81.57 ± 1.22  | 45.47 ± 2.52         | 20.44 ± 3.48         | 56.76     |
|         | FairAdjT2=5        | 88.49 ± 1.30| 90.34 ± 1.36| 80.97 ± 0.83  | 42.39 ± 2.95         | 16.75 ± 3.26         | 70.37     |
|         | FairAdjT2=20       | 85.97 ± 0.62| 87.84 ± 0.53| 77.68 ± 0.54  | 35.67 ± 1.45         | 11.26 ± 3.07         | 145.01    |
|         | CFGE               | 90.76 ± 1.27| 92.59 ± 1.30| 82.35 ± 1.26  | 32.06 ± 1.33         | 13.73 ± 1.63         | 531.79    |
| Citeseer| VGAE               | 88.24 ± 1.83| 90.53 ± 2.78| 81.67 ± 3.39  | 24.82 ± 2.67         | 7.13 ± 3.32          | 20.33     |
|         | FairAdjT2=2        | 87.90 ± 1.81| 89.99 ± 2.81| 81.61 ± 3.02  | 24.17 ± 2.65         | 6.49 ± 3.34          | 84.10     |
|         | FairAdjT2=5        | 87.67 ± 1.74| 89.48 ± 2.67| 81.50 ± 3.01  | 23.65 ± 2.72         | 6.47 ± 3.47          | 109.07    |
|         | FairAdjT2=20       | 86.90 ± 2.23| 88.61 ± 3.23| 80.41 ± 3.35  | 22.74 ± 3.23         | 6.47 ± 3.38          | 240.16    |
|         | CFGE               | 87.56 ± 3.29| 90.45 ± 3.11| 71.12 ± 13.42 | 13.60 ± 4.86         | 3.00 ± 1.39          | 478.18    |
|         | NetMF              | 86.72 ± 0.73| 91.20 ± 0.46| 84.53 ± 0.30  | 23.32 ± 1.48         | 4.66 ± 1.72          | 19.35     |
|         | FairMILE-NetMF     | 89.22 ± 0.34| 91.71 ± 0.69| 85.30 ± 0.58  | 20.96 ± 1.65         | 2.79 ± 1.45          | 7.57      |
|         | DeepWalk           | 88.83 ± 0.26| 90.96 ± 0.31| 81.59 ± 1.54  | 23.99 ± 1.88         | 5.55 ± 3.14          | 48.10     |
|         | FairMILE-DeepWalk  | 89.50 ± 1.02| 92.55 ± 0.70| 86.39 ± 0.37  | 17.67 ± 1.18         | 2.62 ± 1.31          | 21.20     |
|         | Node2vec           | 89.00 ± 0.65| 91.77 ± 0.35| 83.88 ± 0.81  | 22.85 ± 1.57         | 3.52 ± 2.17          | 16.88     |
|         | Fairwalk           | 88.86 ± 0.85| 91.71 ± 0.33| 83.52 ± 0.61  | 23.59 ± 1.38         | 4.13 ± 2.43          | 16.78     |
|         | FairMILE-Node2vec  | 89.54 ± 0.95| 92.31 ± 0.52| 86.91 ± 0.78  | 12.49 ± 0.91         | 2.08 ± 1.08          | 11.72     |
opinions, findings, and conclusions in this material are those of the author(s) and may not reflect the views of the respective funding agency.

REFERENCES

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

[2] Taha Atahan Akyildiz, Amro Alabesi Aljundi, and Kamer Kaya. 2020. GOSH: Embedding big graphs on small hardware. In ICPP. 1–11.

[3] Abdulrazzak Asudeh, HV Jagadish, Julia Stoyanovich, and Gautam Das. 2019. Designing fair ranking schemes. In SIGMOD. 1257–1267.

[4] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam Tauman Kalai. 2016. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. In NeurIPS. Vol. 29.

[5] Awanthika de Silva and William Hamilton. 2019. Compositional fairness constraints for graph embeddings. In International Conference on Machine Learning. PMLR, 715–724.

[6] Haochen Chen, Bryan Perozzi, Yifan Hu, and Steven Skiena. 2018. HARP: hiearchical representation learning for networks. In Proceedings of the 32nd AAAI Conference on Artiﬁcial Intelligence. 2127–2134.

[7] Lee Cohen, Zachary C. Lipton, and Yishay Mansour. 2020. Efﬁcient Candidate Screening Under Multiple Tests and Implications for Fairness. In 1st Symposium on Foundations of Responsible Computing, Foresce 2020. Vol. 156. 1–11.

[8] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering 31, 5 (2018), 833–852.

[9] Yushun Dong, Jian Li, Kuansan Wang, and Jie Tang. 2018. A survey on network embedding. arXiv preprint arXiv:1804.09888 (2018).

[10] Andrew Lohn and Micah Musser. 2022. AI and Compute: How Much Longer Can We Afford. In Proceedings of the 13th Innovations in Theoretical Computer Science Conference. 214–226.

[11] Ashudeep Singh and Thorsten Joachims. 2018. Fairness of exposure in rankings. In Proceedings of the 29th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 701–710.

[12] Indro Spinelli, Simone Scardapane, Amir Hussain, and Aurelio Uncini. 2021. FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning. In Proceedings of the ICLR Conference on Learning Representations.