Rethinking Fair and Stable Representation Learning

Prasoon Sinha [1] **Zulfidin (Bobojon) Khodzhaev** [1] **Yizhou Wang** [1] **Vijay Lingam** [1]

Abstract

Machine Learning, particularly in the context of Graph Neural Networks (GNNs), is being deployed across various applications, including safety-critical domains. While the accuracy of these systems remains a critical consideration, it is equally important to examine them through the lenses of fairness and stability. Prior research has introduced a unified framework aimed at learning stable and fair node representations. In this work, we revisit and refine this framework, presenting a simplified version that demonstrates superior performance in terms of both accuracy and fairness/stability metrics. Furthermore, we introduce a set of straightforward baseline methods that prove to be effective and competitively perform, often surpassing the performance of previous state-of-the-art (SOTA) approaches. The code for our work is available at https://tinyurl.com/AMLFinalCode.

1. Introduction

Machine Learning (ML) based applications are becoming increasingly ubiquitous and popular, and are being deployed in safety-critical environments ranging from healthcare and drug design to law regulation (Caton & Haas, 2023). Representation Learning, which is at the core of ML, manifests itself in various forms depending on the kind of downstream task at hand. It therefore becomes vital to ensure that the representations learned by learning algorithms are safe and reliable. More importantly, ML models should not exhibit discriminatory biases (e.g., invariance to sensitive attributes) and should be robust to minor perturbations to these sensitive features – the output should not change drastically with minor changes to these attributes.

Previous works (Zemel et al., 2013; Ma et al., 2022) have explored fairness (specifically, counterfactual fairness) and robustness in representation learning independently. A more recent work by Agarwal et al. (2021) introduces a framework, NIFTY, which unifies both fairness and stability by proposing a new model architecture and explicit objective functions to govern them. However, these approaches are complex and do not offer comparison against classic ML models. We address these shortcomings by offering a comprehensive comparison against several simple yet effective baselines and propose a new model that is at least as effective as NIFTY. We enumerate our contributions below.

1. Offer a comprehensive evaluation over a suite of models ranging from simple classic ML models (Logistic Regression, MLP, other off-the-shelf classifiers) to Graph Representation models (like GCN (Kipf & Welling, 2017)).

2. Explore the role of dimensionality reduction techniques in mitigating discriminatory biases in representation learning.

3. Propose a simple regularization term that can be appended to any Graph Neural Network model that is at least as effective as NIFTY, current state-of-the-art fairness and robustness baseline.

2. Related Work

Our work can be placed at the intersection of fairness, robustness, representation learning, and Graph Neural Networks (GNNs). We briefly discuss each of these domains in the context of our work.

Fairness in ML. This is an emerging field in ML that aims to study ways to avoid model and data biases (Oneto & Chiappa, 2020). Fairness can be broadly categorized into 1) *group-fairness*: minority groups receive similar treatment as advantaged groups (Hardt et al., 2016b), 2) *individual fairness*, which encapsulates that similar individuals receive similar treatment (Dwork et al., 2012) and 3) *counterfactual fairness*, which deems a decision pertaining to an individual as fair if it is invariable to their sensitive attribute (Kusner et al., 2017). In other words, perturbing the sensitive attribute does not lead to a change in decision. Different metrics were proposed to quantify each of these fairness definitions. We describe the metrics we use in our work in Section 4.1.

GNNs. GNNs are a popular class of methods for semi-supervised graph learning tasks including node classification, link prediction and graph classification. GNNs exploit the topological structure in addition to the node feature

attributes and optional edge attributes to learn node representations that are useful for downstream tasks. GNN approaches at a high level either use different aggregation schemes (Kipf & Welling, 2017; Veličković et al., 2018) or use derived graph filters (He et al., 2021) to learn node representations. In this work, we restrict our scope to Graph Convolutional Networks (GCNs), one of the popular GNN architectures. However, our proposed approach can also accommodate other GNNs.

Fairness and Robustness. Recent studies (Dai & Wang, 2021; Zhu et al., 2019) have independently delved into the realms of fairness and robustness within GNNs. These investigations propose specialized loss functions or employ adversarial training techniques to mitigate bias and fortify models against perturbations in input features. A more recent contribution by Agarwal et al. (2021) consolidates these approaches into a unified framework. However, this framework necessitates multiple augmented views of the input graph with perturbations and dual optimizers to effectively learn both fair and stable node representations.

In our current study, we aim to devise a simpler strategy that does not compromise significantly on performance, in addressing the challenges of fairness and robustness.

3. Evaluating ML Models With Sensitive Attributes

Although GNNs have become popular in deployed real-world applications, in this project we take a step back and evaluate the efficacy and potential discriminatory bias of simple ML classifiers in addition to the state-of-the-art graph-based techniques. With both types of models, we will also explore the role of dimensionality reduction techniques in mitigating bias. For example, we will use PCA and t-SNE to capture the most variance in the dataset and determine whether reducing the dimensionality essentially removes the ability for our models to become biased during training.

3.1. Graph Based Approaches

Since our evaluation datasets contain a graph structure associated with them, Graph Neural Networks (GNNs) based approaches would be an ideal paradigm for learning representations required to attempt the downstream task. Several recent works (Dai & Wang, 2021; Agarwal et al., 2021) were proposed to make GNNs fair and robust against discriminative biases and input perturbations.

One of the popular GNNs that these works build on is a Graph Convolutional Network (GCN) (Kipf & Welling, 2017). We briefly describe the technical details of a GCN below.

3.1.1. Graph Convolutional Network

A graph convolutional network is a multi-layer network architecture with the first (*aka* input) layer taking the node feature matrix ($X \in \mathcal{R}^{n \times m}$ where n and m denote the number of nodes and input feature dimension) as input. With $H(0) = X$, each layer consumes its previous layer's output (i.e., embeddings of all nodes in the graph, denoted as $H(k-1)$) and produces a new embedding matrix ($H(k) \in \mathcal{R}^{n \times d}$) where d is the embedding dimension). Thus, the core convolutional layer is specified as:

$$H(k) = \sigma(\tilde{S}H(k-1)W_k) \tag{1}$$

where W_k and σ denote k^{th} layer weight matrix and nonlinear activation function (e.g., a rectilinear activation function ReLU$(\cdot)$). $\tilde{S}$ refers to the symmetric normalized adjacency graph.

Our main graph-based baselines would be GCN (Kipf & Welling, 2017), NIFTY+GCN, and FAIRGNN (Dai & Wang, 2021).

3.2. Non-graph Based Approaches

For completeness, we also evaluate against several simple ML classifiers. The goal of this portion of the work is to (1) evaluate how robust these classifiers are to sensitive attribute(s), and (2) evaluate if they can be as effective in their predictions as GNNs while being more robust. Our baselines include Logistic Regression (LR), MLP and CatBoost. LR incorporates L1 and L2 regularization to prevent overfitting and help with feature selection. Decision trees offer interpretable predictions and natively handle categorical features. MLPs provide flexibility to design custom loss functions and representations.

To confront the challenges of high-dimensional data, dimensionality reduction techniques such as PCA and t-SNE were implemented in those non-graph approaches. PCA was used to transform the feature space into principal components, thereby reducing complexity while preserving critical information. In contrast, t-SNE was employed to retain the local structure of data in a reduced-dimensional space, facilitating the interpretation of complex datasets.

4. Datasets and Experimental Setup

This study uses three publicly accessible graph-structured datasets made available by Agarwal et al. (2021). The first, a **German Credit Dataset**, contains 1,000 samples of clients. Each sample has 29 features, where most of the features are details about the client's finances while a few features are characteristics of the client, like their age, gender, and marital status. Clients are connected in a graph structure based on their similarity in financial history. The goal is to classify clients into good and bad credit risks.

Table 1. Dataset information.

Dataset	German Credit Dataset	Recidivism Dataset	Credit Defaulter Dataset
# Nodes	1,000	18,876	30,000
# Edges	22,242	321,308	1,436,858
# Node Features	29	18	13
Sensitive Attribute	Gender	Race	Age
Node Labels	Good Credit v.s. Bad Credit	Violent v.s. Nonviolent Crime	Payment Default v.s. No Default

The second, **Recidivism Dataset**, comprises 18,876 samples of defendants who were released on bail to predict future violent behavior among defendants. It contains demographic attributes and past criminal records as features. Defendants are connected based on the similarity of past criminal records and demographics.

The third, **Credit Defaulter Dataset**, contains 30,000 data points of credit card holders with features capturing education, credit history, age, and spending/payment details. Credit card holders are connected based on their similarity in spending patterns. The goal is to predict whether each credit card holder will default on their next payment.

In all three datasets shown in Table 1, the similarity measure between two samples is constructed using the Minkowski distance: 1/(1 + minkowski(x_u, x_v)). In the German Credit graph, two nodes are connected if the similarity is 80% of the maximum similarity between all respective nodes. Similarly, the similarity threshold for edge connections is 60% and 70% for the Recidivism and Credit Defaulter graphs respectively. These datasets allow for in-depth analysis of algorithmic fairness by modeling relationships through graph structures.

4.1. Evaluation Metrics

To measure the predictive performance on the downstream tasks, we will report the AUROC score and F1 score.

To quantify the group fairness of the model, we will report statistical parity (SP) (Dwork et al., 2011) defined as:

$$\Delta_{SP} = |P(\hat{y_u} = 1|s = 0) - P(\hat{y_u} = 1|s = 1)| \quad (2)$$

and equal opportunity (EO) (Hardt et al., 2016a) defined as:

$$\begin{aligned}\Delta_{EO} = |P(\hat{y_u} = 1|y_u = 1, s = 0) \\ -P(\hat{y_u} = 1|y_u = 1, s = 1)|\end{aligned} \quad (3)$$

where u represents some node in the graph and s is the value of the sensitive feature.

To measure counterfactual fairness, we will use the unfairness score representing the influence of sensitive features on the prediction and the instability score showing how much the prediction is affected by random noise to features (Agarwal et al., 2021).

5. Proposed Approach

We start by recapping the core components of the NIFTY framework. As a first step, NIFTY applies a spectral normalization to all the weight matrices in the chosen GNN model. Next, two augmented views of the input graph where sensitive attributes and node features are independently perturbed are created. Finally, each node features in the original and augmented graphs are projected to the same dimension vectors using an encoder, and a triplet similarity loss is applied. Please refer to Agarwal et al. (2021) for several other implementation-level details.

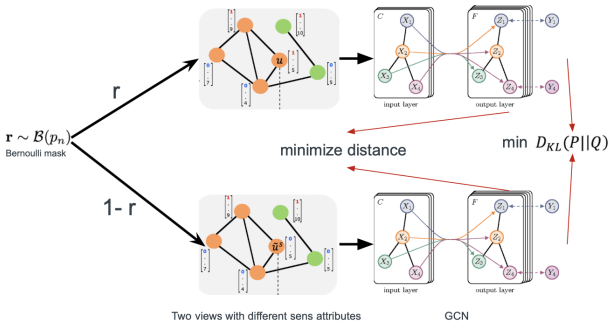

Figure 1. Illustration of our proposed approach.

Our ablative studies reveal that the major contributor to stability is the spectral norm. Intuitively, spectral norm ensures that all the layer weights are in a similar space, and are therefore more robust to minor input perturbations. For theoretical analysis, please refer to Section 5 of Agarwal et al. (2021).

We simplify the NIFTY framework, by reducing the two augmented graphs into a single augmented graph. In specific, in every training step, we randomly perturb training

node features, graph edges, and sensitive attributes by drawing a binary vector from Bernoulli distribution ($r \sim \beta(p_i)$) and masking each attribute respectively. We then propose to minimize the distance between the intermediate and final layer representations using original and augmented graphs. The final loss can be written as:

$$\mathcal{L} = \mathcal{L}_c + \lambda \mathbb{E}_{u \sim D_{train}}[dist(z_u, z_{\hat{u}}) + KL(y_u, y_{\hat{u}})] \quad (4)$$

In the above equation $\mathcal{L}_c$ corresponds to binary cross-entropy loss, λ is the weight assigned to the regularization term, D_{train} refers to the training nodes, $z_u, z_{\hat{u}}$ refer to original and augmented (perturbed) graph node u's latent representation and y refers to output probabilities. We try both euclidean and cosine-distance as options for $dist$ and KL refers to Kullback–Leibler divergence. We illustrate our approach in Figure 1.

6. Results

We present our experimental results for all the three baseline datasets in Table 2. We report mean performance along with standard deviation across 5 runs with different random seeds. Due to space constraints, we only report results for the best baseline model variants. We make all results available in the appendix.

6.1. Graph based approach

We compare our proposed approach against NIFTY and tabulate our results in Table 2. We can infer that our approach retains AUROC and F1 scores close to that of GCN and achieves a perfect unfairness score of 0. Our approach offers competitive or better performance on other fairness and stability metrics. We see similar observations for the remaining datasets. We tabulate these results in Appendix A.

6.2. Non-graph based approach: LogReg

Logistic regression (LogReg) is one of the classic ML models used in our project. The goal was to ensure accurate, interpretable predictions while avoiding the complexity and computational demands of more elaborate models. The LogReg models were configured with various regularization strengths *C* from 0.01 to 10000 and diverse solvers to use in the optimization problem including `'lbfgs'`, `'liblinear'`, `'newton-cg'`, `'sag'`, and `'saga'`. We introduced those additional hyperparameters in order to consider different regularization possibilities, helping in fine-tuning the logistic regression model to prevent overfitting and finding the right balance between model simplicity and the ability to capture complex patterns in the data.

To select features, Principal Component Analysis (PCA) was utilized for each dataset, converting the feature space into a series of principal components that are linearly uncorrelated. The PCA was executed on the training data, employing a range of hyperparameters, including various Singular Value Decomposition (SVD) solvers such as `'auto'`, `'full'`, `'randomized'`, and `'arpack'`. In addition, the number of components selected varied from a single component to the total number of features in the dataset (or one less than the total feature count when using `'arpack'` as the SVD solver). This approach facilitated the examination of the effects of different extents of dimensionality reduction. Post the PCA fitting, both the training and test datasets were transformed based on these principal component representations. By experimenting with diverse hyperparameter settings, the most effective values were identified in relation to the accuracy of the test data.

For t-SNE, another dimensionality reduction method, we capped the component count at 3 to balance computational efficiency and ease of interpretation. This process entailed fitting t-SNE to the entire dataset and then applying the transformation to both the training and testing sets. For practical computational considerations, the 'Barnes Hut' approximation method was used in the t-SNE reduction. Similar to the approach taken with PCA, we trained and tested the logistic regression classifier on the data reduced via t-SNE to evaluate the effect of this dimensionality reduction technique on the model's performance to determine the optimal number of components based on test accuracy.

The study highlighted significant variability in the efficacy of logistic regression models depending on the dataset and hyperparameter configurations used. Specifically, while the AUROC and F1 scores declined for the German Credit Dataset when PCA or t-SNE was applied, the Recidivism Dataset saw an improvement in performance with PCA, where the AUROC score rose from 95.39% to 95.73%, and the F1 score increased from 89.61% to 90.08%. In contrast, these metrics for the Credit Defaulter Dataset remained relatively stable with PCA, registering 74.47% for AUROC and 81.81%∼81.80% for F1 score, underscoring the distinct behaviors of different datasets.

Different feature selection techniques also impact the levels of unfairness and instability in distinct ways. For instance, the unfairness score in the German Credit Dataset was reduced from 4.80% to 2.56% using PCA and to 0.24% with t-SNE. However, only PCA was effective in lowering the unfairness score in the other two datasets. In addition, the instability score in the German Credit Dataset dropped from 32.48% to 30.80% with PCA and to a mere 0.16% with t-SNE, but t-SNE alone was successful in reducing the instability score by 5.19% in the Recidivism Dataset, and for the Credit Defaulter Dataset, only PCA was able to prevent a worsening of the instability score.

Table 2. Best results comparing baselines and proposed approaches on the three datasets. We report mean performance along with standard deviation across 5 runs with different seeds.

Dataset	Method	AUROC (↑)	F1-Score (↑)	Unfairness (↓)	Instability (↓)	$\Delta_{SP}(\downarrow)$	$\Delta_{EO}(\downarrow)$
German Credit Dataset	GCN + SIM LOSS (ours)	73.74 (1.40)	73.74 (1.40))	**0.00 (0.00)**	5.32 (3.56)	20.14 (4.97)	4.73 (3.11)
	NIFTY-GCN	68.56 (4.12)	65.15 (32.02)	0.48 (0.64)	1.36 (1.00)	8.23 (7.84)	7.25 (5.92)
	LOGREG T-SNE	55.41 (0.00)	62.58 (0.10)	0.24 (0.32)	**0.16 (0.32)**	**0.89 (0.09)**	**1.91 (0.67)**
	MLP	**76.02 (1.01)**	80.28 (1.88)	14.64 (14.00)	28.64 (5.36)	28.49 (3.21)	17.65 (2.95)
	MLP PCA	49.36 (5.03)	**82.50 (0.34)**	42.56 (37.83)	44.88 (38.39)	7.21 (5.68)	7.25 (6.85)
Recidivism Dataset	GCN + SIM LOSS (ours)	86.49 (0.64)	77.60 (0.32)	**0.00 (0.00)**	**7.28 (1.04)**	2.14 (1.34)	3.75 (0.72)
	NIFTY-GCN	81.40 (0.89)	69.24 (0.70)	0.84 (0.68)	13.28 (1.62)	3.16 (0.60)	**2.99 (0.40)**
	LOGREG PCA	95.73 (0.00)	90.08 (0.00)	**0.00 (0.00)**	43.10 (2.11)	7.60 (0.00)	6.85 (0.00)
	MLP T-SNE	51.09 (1.22)	26.31 (18.44)	41.78 (8.27)	46.98 (1.66)	**1.60 (1.58)**	3.32 (3.02)
	CATBOOST	**98.97 (0.13)**	**98.59 (0.82)**	2.58 (0.42)	44.06 (2.31)	6.45 (1.22)	5.96 (0.64)
Credit Defaulter Dataset	GCN + SIM-LOSS (ours)	73.39 (0.62)	**84.58 (2.49)**	**0.00 (0.00)**	0.87 (0.87)	**7.69 (1.19)**	**5.92 (2.06)**
	NIFTY-GCN	72.07 (0.17)	81.75 (0.05)	0.03 (0.03)	**0.35 (0.13)**	12.72 (1.62)	10.09 (1.55)
	LOGREG PCA	74.47 (0.00)	81.80 (0.00)	0.01 (0.00)	16.56 (0.39)	11.81 (0.00)	9.67 (0.00)
	MLP T-SNE*	71.90 (0.48)	80.09 (2.71)	10.03 (3.34)	32.25 (7.37)	16.30 (3.71)	14.45 (3.95)
	CATBOOST	**76.39 (0.32)**	82.26 (1.34)	0.04 (0.00)	17.08 (0.00)	12.28 (1.46)	9.80 (2.24)

* Results obtained after randomly flipping sensitive attributes to augment the data.

On the other hand, both dimensionality reduction techniques improved the statistical parity (Δ_{SP}), and equal opportunity (Δ_{EO}) metrics in most scenarios except in the instance of using PCA as the feature selection technique for the Recidivism Dataset. In this particular case, Δ_{EO} saw a slight increase, going from 6.61% to 6.85%.

6.3. Non-graph based approach: MLP

Multi-layer perceptrons (MLPs) were selected for their versatility in handling diverse data types and intricate relationships. A strategic choice was made to utilize 50 and 100 nodes in the hidden layers of the MLPs to strike a balance between model complexity and computational efficiency. This decision was informed by the feature sizes of the datasets: the German Credit Dataset with 27 features, the Credit Defaulter Dataset with 13 features, and the Bail Dataset with 18 features. These configurations were designed to effectively capture patterns in datasets varying in feature count and sample size, ranging from 1,000 to 30,000 entries. The objective was to avert overfitting, especially in datasets with fewer features, while assessing the impact of various network architectures on performance.

The MLPs were configured with diverse hidden layer sizes and activation functions, including `'identity'`, `'logistic'`, `'tanh'`, and `'relu'`. This range, from simpler single-layered networks to more complex structures like (50, 50) and (100, 100) multi-layered networks, set the stage for empirical experimentation and adjustments based on performance.

The introduction of additional hyperparameters like learning rate, batch size, and regularization parameters was intended to allow more refined tuning of neural network training. However, this added complexity also made it more difficult to maintain consistent performance across experiments. Specifically, changes to these parameters can substantially alter the training dynamics, leading neural networks to converge differently and develop different generalization capabilities. To reduce this source of variability, this study used the default library settings for the learning rate, batch size, and regularization parameters.

The dimensionality reduction techniques of PCA and t-SNE utilized in this study followed the same methodology and implementation described in the previous study in Section 6.2.

Performance varied significantly across models and datasets. On the German Credit Dataset, the base MLP achieved 76.02% AUROC, 80.28% F1 but concerning fairness deviations (Δ_{SP} 28.49%, Δ_{EO} 17.65%). Perturbations improved parity/opportunity but reduced accuracy. PCA performed poorly overall, while t-SNE achieved near metric parity despite lower predictive capability.

On the Credit Defaulter Dataset, base MLP scored well on accuracy (74.92% AUROC, 83.50% F1) and fairness. MLP* maintained strong performance with slight fairness gains.

However, the Recidivism base MLP saw high predictive power (94.42% AUROC, 88.43% F1) decline substantially in MLP* on both accuracy and fairness treatment.

No approach reliably balanced all objectives across datasets. Trade-offs emerged between accuracy, fairness, and robustness to changes. The sensitivity of configurations across data distributions underscores the need for adaptable, reliable techniques that can achieve multifaceted metrics on diverse real-world datasets. Progress remains through meticulous analysis and advanced flexible architectures.

6.4. Non-graphed based approach: CatBoost

We finally evaluated the efficacy of boosting. Boosting enhances predictive accuracy by combining weak models in an iterative manner. Upon misclassification, it places more importance on misclassified instances, progressively improving model performance with each iteration. We opted to use the popular CatBoost algorithm for its strength in handling categorical features by employing ordered boosting to improve its optimizations in handling categorical features.

We performed a hyperparameter search on various combinations of iterations, learning rate, and depth hyperparameters. We evaluated values of iterations from 100 to 1500 (affecting the number of iterations or trees created), learning rate from 0.01 to 0.1, and depth (tree depth) from 4 to 10. The goal of the hyperparameter search was to ensure the models make accurate predictions, but do not overfit to the training dataset. Across our three datasets, we noticed less complex models performed better, with a lower number of trees (200-600) and depths (5-7). Meanwhile, the best learning rate tended to be a little larger around 0.06 to 0.08. Finally, we note that we perform the reduction technique (PCA and t-SNE) following the same methodology as the previous two studies.

Generally, we see that CatBoost performance follows similar trends to Logistic Regression and MLP: significance variability in the efficacy of CatBoost models depends on the dataset used. While CatBoost produced high AUROC scores of 98.97% and 76.39% for the Recidivism and Credit Defaulter datasets, respectively, its performance was poor with the German Credit Dataset (69.34%). Similar trends hold for the F1 Score.

Unlike logistic regression, only t-SNE had a positive impact on statistical parity and equal opportunity throughout the datasets. For example, the German Credit Dataset saw a $\sim 8\times$ drop in equal opportunity when using t-SNE (compared to plain CatBoost). Moreover, the statistical parity dropped by $\sim 2\times$ in the Recidivism Dataset. Meanwhile, PCA did not seem to help any of the fairness metrics. We generally saw unfairness and instability values rise when using PCA.

The differences in fairness metrics results between logistic regression and CatBoost when using dimensionality reduction techniques show that there is no off-the-shelf approach that overcomes bias in different domains. Ultimately, while these simple models may be useful, they need to be heavily tuned to the dataset.

7. Limitations & Conclusion

In this work, we reexamine the concepts of fairness and stability in representation learning, emphasizing their application to Graph Neural Networks (GNNs). We conduct a thorough comparison with classic Machine Learning (ML) methods, incorporating minor extensions for a comprehensive evaluation. Interestingly, our results reveal that these straightforward baseline methods outperform recent state-of-the-art (SOTA) approaches, including NIFTY. Additionally, we introduce a GNN-based approach that simplifies NIFTY framework while maintaining comparable or even superior performance.

Our experimentation is limited to three datasets introduced by Agarwal et al. (2021). However, for future work, we plan to expand our analysis to include larger graph datasets that contain multiple sensitive attributes.

References

Agarwal, C., Lakkaraju, H., and Zitnik, M. Towards a unified framework for fair and stable graph representation learning. *CoRR*, abs/2102.13186, 2021. URL https://arxiv.org/abs/2102.13186.

Caton, S. and Haas, C. Fairness in machine learning: A survey. *ACM Comput. Surv.*, aug 2023. ISSN 0360-0300. doi: 10.1145/3616865. URL https://doi.org/10.1145/3616865.

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

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. Fairness through awareness. *CoRR*, abs/1104.3913, 2011. URL http://arxiv.org/abs/1104.3913.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pp. 214–226, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450311151. doi: 10.1145/2090236.2090255. URL https://doi.org/10.1145/2090236.2090255.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016a. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pp. 3323–3331, Red Hook, NY, USA, 2016b. Curran Associates Inc. ISBN 9781510838819.

He, M., Wei, Z., Huang, Z., and Xu, H. Bernnet: Learning arbitrary graph spectral filters via bernstein approximation. In *NeurIPS*, 2021.

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

Kusner, M., Loftus, J., Russell, C., and Silva, R. Counterfactual fairness. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 4069–4079, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ma, J., Guo, R., Wan, M., Yang, L., Zhang, A., and Li, J. Learning fair node representations with graph counterfactual fairness. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, pp. 695–703, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3498391. URL https://doi.org/10.1145/3488560.3498391.

Oneto, L. and Chiappa, S. *Fairness in Machine Learning*, pp. 155–196. Springer International Publishing, 2020. ISBN 9783030438838. doi: 10.1007/978-3-030-43883-8_7. URL http://dx.doi.org/10.1007/978-3-030-43883-8_7.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks. *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Zemel, R., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 325–333, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/zemel13.html.

Zhu, D., Zhang, Z., Cui, P., and Zhu, W. Robust graph convolutional networks against adversarial attacks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, pp. 1399–1407, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330851. URL https://doi.org/10.1145/3292500.3330851.

A. Additional Experimental Results

Table 3. Results comparing baselines and proposed approaches on German Credit Dataset. We report mean performance along with standard deviation across 5 runs with different seeds.

Method	**AUROC** (↑)	**F1-Score** (↑)	**Unfairness** (↓)	**Instability** (↓)	$\Delta_{SP}(\downarrow)$	$\Delta_{EO}(\downarrow)$
GCN	72.58 (1.56)	79.30 (1.52)	17.92 (4.34)	11.28 (1.74)	39.56 (4.79)	32.06 (5.59)
GCN + SIM LOSS (ours)	73.74 (1.40)	73.74 (1.40))	**0.00 (0.00)**	5.32 (3.56)	20.14 (4.97)	4.73 (3.11)
NIFTY-GCN	68.56 (4.12)	65.15 (32.02)	0.48 (0.64)	1.36 (1.00)	8.23 (7.84)	7.25 (5.92)
LOGREG	73.34 (0.00)	78.42 (0.00)	4.80 (0.00)	32.48 (2.08)	30.73 (0.00)	19.22 (0.00)
LOGREG PCA	72.05 (0.52)	76.15 (0.33)	2.56 (1.51)	30.80 (5.95)	29.36 (4.59)	18.72 (3.73)
LOGREG T-SNE	55.41 (0.00)	62.58 (0.10)	0.24 (0.32)	**0.16 (0.32)**	**0.89 (0.09)**	**1.91 (0.67)**
MLP	**76.02 (1.01)**	80.28 (1.88)	14.64 (14.00)	28.64 (5.36)	28.49 (3.21)	17.65 (2.95)
MLP PCA	49.36 (5.03)	**82.50 (0.34)**	42.56 (37.83)	44.88 (38.39)	7.21 (5.68)	7.25 (6.85)
MLP T-SNE	55.43 (0.12)	71.12 (7.50)	38.48 (8.77)	39.36 (8.30)	1.12 (1.20)	2.50 (0.94)
MLP*	73.60 (3.79)	72.45 (7.11)	16.64 (15.08)	29.20 (6.61)	21.07 (9.13)	13.05 (7.06)
MLP PCA*	53.15 (9.07)	72.36 (16.76)	30.72 (25.70)	18.24 (8.18)	7.31 (5.49)	7.44 (5.96)
MLP T-SNE*	55.47 (0.11)	71.08 (7.25)	38.00 (9.08)	37.92 (9.96)	1.09 (1.23)	3.00 (1.35)
CATBOOST	69.34 (4.52)	75.26 (3.59)	5.28 (2.42)	32.80 (0.00)	32.65 (0.00)	24.58 (0.00)
CATBOOST PCA	74.23 (2.43)	78.54 (2.84)	7.86 (1.53)	24.93 (4.42)	28.84 (1.84)	14.82 (2.95)
CATBOOST T-SNE	60.09 (0.24)	60.54 (0.28)	8.48 (1.72)	9.54 (5.34)	4.36 (2.83)	3.62 (0.35)

* Results obtained after randomly flipping sensitive attributes to augment the data.

Table 4. Results comparing baselines and proposed approaches on the Recidivism Dataset. We report mean performance along with standard deviation across 5 runs with different seeds.

Method	**AUROC** (↑)	**F1-Score** (↑)	**Unfairness** (↓)	**Instability** (↓)	$\Delta_{SP}(\downarrow)$	$\Delta_{EO}(\downarrow)$
GCN	86.52 (0.42)	77.50 (0.87)	9.02 (3.04)	21.97 (1.63)	8.49 (0.73)	5.93 (0.56)
GCN + SIM LOSS (ours)	86.49 (0.64)	77.60 (0.32)	**0.00 (0.00)**	**7.28 (1.04)**	2.14 (1.34)	3.75 (0.72)
NIFTY-GCN	81.40 (0.89)	69.24 (0.70)	0.84 (0.68)	13.28 (1.62)	3.16 (0.60)	**2.99 (0.40)**
LOGREG	95.39 (0.00)	89.61 (0.00)	0.06 (0.00)	42.53 (2.55)	7.52 (0.00)	6.61 (0.00)
LOGREG PCA	95.73 (0.00)	90.08 (0.00)	**0.00 (0.00)**	43.10 (2.11)	7.60 (0.00)	6.85 (0.00)
LOGREG T-SNE	54.15 (0.09)	42.50 (0.77)	2.61 (0.39)	37.34 (8.75)	3.88 (0.21)	6.54 (0.26)
MLP	94.42 (0.18)	88.43 (0.27)	0.45 (0.12)	42.32 (2.42)	2.42 (0.28)	3.57 (0.58)
MLP PCA	96.02 (0.04)	90.90 (0.03)	2.31 (0.29)	42.64 (2.67)	6.88 (0.20)	7.43 (0.09)
MLP T-SNE	51.09 (1.22)	26.31 (18.44)	41.78 (8.27)	46.98 (1.66)	**1.60 (1.58)**	3.32 (3.02)
MLP*	95.75 (0.25)	90.59 (0.30)	2.51 (0.68)	45.81 (2.27)	7.12 (2.14)	5.68 (3.99)
MLP PCA*	96.30 (0.06)	91.60 (0.26)	3.38 (0.53)	45.86 (2.04)	5.93 (1.95)	5.24 (2.53)
MLP T-SNE*	51.96 (1.08)	31.43 (15.89)	39.70 (4.75)	46.12 (2.09)	3.02 (1.70)	4.23 (2.17)
CATBOOST	**98.97 (0.13)**	**98.59 (0.82)**	2.58 (0.42)	44.06 (2.31)	6.45 (1.22)	5.96 (0.64)
CATBOOST PCA	91.97 (0.38)	88.24 (0.26)	6.38 (0.82)	44.84 (1.38)	8.25 (1.58)	5.82 (0.47)
CATBOOST T-SNE	53.04 (2.84)	44.17 (2.27)	19.75 (0.27)	47.07 (1.53)	3.28 (2.12)	5.93 (2.58)

* Results obtained after randomly flipping sensitive attributes to augment the data.

Table 5. Results comparing baselines and proposed approaches on the Credit Defaulter Dataset. We report mean performance along with standard deviation across 5 runs with different seeds.

Method	**AUROC** (↑)	**F1-Score** (↑)	**Unfairness** (↓)	**Instability** (↓)	$\Delta_{SP}(\downarrow)$	$\Delta_{EO}(\downarrow)$
GCN	70.15 (7.40)	82.19 (0.58)	2.47 (2.24)	7.04 (2.10)	11.72 (2.21)	9.88 (1.40)
GCN + SIM-LOSS (ours)	73.39 (0.62)	**84.58 (2.49)**	**0.00 (0.00)**	0.87 (0.87)	**7.69 (1.19)**	**5.92 (2.06)**
NIFTY-GCN	72.07 (0.17)	81.75 (0.05)	0.03 (0.03)	**0.35 (0.13)**	12.72 (1.62)	10.09 (1.55)
LOGREG	74.47 (0.00)	81.81 (0.00)	0.05 (0.00)	16.57 (0.39)	11.84 (0.00)	9.69 (0.00)
LOGREG PCA	74.47 (0.00)	81.80 (0.00)	0.01 (0.00)	16.56 (0.39)	11.81 (0.00)	9.67 (0.00)
LOGREG T-SNE	71.33 (0.11)	76.10 (0.23)	4.46 (0.90)	24.14 (1.01)	10.58 (0.97)	8.49 (1.04)
MLP	74.92 (0.12)	83.50 (0.17)	1.24 (1.39)	34.10 (3.23)	15.78 (0.53)	13.38 (0.75)
MLP PCA	74.16 (0.36)	82.91 (1.20)	5.45 (1.99)	34.83 (1.01)	14.07 (2.58)	11.49 (2.67)
MLP T-SNE	72.53 (0.78)	79.83 (1.10)	8.43 (2.99)	31.85 (7.65)	16.72 (4.22)	15.14 (4.80)
MLP*	74.89 (0.20)	82.90 (0.96)	2.68 (1.88)	33.71 (1.17)	13.90 (0.83)	11.34 (0.98)
MLP PCA*	73.72 (0.15)	82.40 (1.06)	6.38 (2.45)	35.38 (0.68)	11.23 (2.77)	8.47 (3.29)
MLP T-SNE*	71.90 (0.48)	80.09 (2.71)	10.03 (3.34)	32.25 (7.37)	16.30 (3.71)	14.45 (3.95)
CATBOOST	**76.39 (0.32)**	82.26 (1.34)	0.04 (0.00)	17.08 (0.00)	12.28 (1.46)	9.80 (2.24)
CATBOOST PCA	73.82 (0.43)	83.01 (1.38)	4.36 (1.58)	23.58 (8.27)	13.54 (3.82)	10.83 (1.39)
CATBOOST T-SNE	73.13 (0.48)	82.60 (1.86)	8.28 (1.83)	26.83 (4.98)	14.28 (2.82)	11.74 (3.92)

* Results obtained after randomly flipping sensitive attributes to augment the data.