Towards a Unified Framework for Fair and Stable Graph Representation Learning

Anonymous Authors[1]

Abstract

As the representations output by Graph Neural Networks (GNNs) are increasingly employed in real-world applications, it becomes important to ensure that these representations are fair and stable. In this work, we establish a key connection between fairness and stability and leverage it to propose a novel framework, NIFTY (uNIfying Fairness and stabiliTY), which can be used with any GNN to learn fair and stable representations. We introduce an objective function that simultaneously accounts for fairness and stability and proposes layer-wise weight normalization of GNNs using the Lipschitz constant. Further, we theoretically show that our layer-wise weight normalization promotes fairness and stability in the resulting representations. We introduce three new graph datasets comprising of high-stakes decisions in criminal justice and financial lending domains. Extensive experimentation with the above datasets demonstrates the efficacy of our framework.

1. Introduction

Over the past decade, there has been a surge of interest in leveraging GNNs for graph representation learning. GNNs are used to learn powerful representations for downstream applications—*e.g.,* predicting protein-protein interactions (Huang et al., 2020), drug repurposing (Zitnik et al., 2018), crime forecasting (Jin et al., 2020). As GNNs are increasingly implemented in real-world applications, it becomes important to ensure that these models and their representations are safe and reliable. Specifically, ensuring that the model's resulting representations are not perpetrating undesirable discriminatory biases (*i.e.,* fairness), and are robust to attacks resulting from small perturbations to the graph structure and node attributes (*i.e.,* stability).

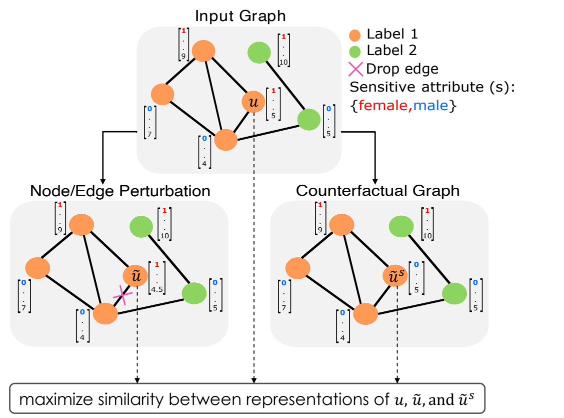

Figure 1. NIFTY can learn fair and stable representations (*i.e.,* invariant to the sensitive attribute value and perturbations to the graph structure and non-sensitive attributes) by maximizing the similarity between representations from diverse augmented graphs.

A myriad of GNN methods with various neighborhood aggregation schemes have been developed recently (*e.g.,* Kipf & Welling (2017); Hamilton et al. (2017); Xu et al. (2018; 2019); Veličković et al. (2019)). While these methods achieve state-of-the-art performance in tasks such as node classification and link prediction, they can be prone to discrimination and instability (Dai & Wang, 2021; Rahman et al., 2019; Bose & Hamilton, 2019). Since GNNs compute node representations by propagating and aggregating neural messages along edges in graph neighborhoods, nodes with similar sensitive attribute values are likely to share similar representations leading to severe discriminatory biases. While previous techniques study fairness (Dai & Wang, 2021) and stability (Zhu et al., 2019) independently, it remains an open question whether there is any deeper connection between these properties, and if they can be achieved simultaneously.

Present work. To tackle the problem of learning fair and stable representations, we first identify a key connection between fairness and stability. While stability accounts for robustness *w.r.t.* random perturbations to node attributes and/or edges, fairness accounts for robustness *w.r.t.* modifications of the sensitive attribute. We use the above connection to develop NIFTY to enforce fairness and stability in the objective function as well as in the message passing step of the GNN layers. Results show that NIFTY improves the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

fairness and stability of five GNNs by 92.01% and 60.87% without sacrificing their predictive performances.

2. Preliminaries

Let $\mathcal{G}=(\mathcal{V}, \mathcal{E}, \mathbf{X})$ denote an undirected graph on nodes $\mathcal{V}$ and= edges $\mathcal{E}$. Let $\mathbf{X}=\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ denote vectors corresponding to the nodes in $\mathcal{V}$, where $\mathbf{x}_v \in \mathbb{R}^M$ captures node v's attributes. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be the adjacency matrix where $\mathbf{A}_{uv}=1$ if there exists some edge between nodes u and v, and otherwise 0. We use $\mathcal{N}_u$ to denote the immediate neighbors of node u, *i.e.,* $\mathcal{N}_u=\{v \in \mathcal{V} | A_{uv}=1\}$. Furthermore, let $\mathbf{I}_u \in \{0,1\}^N$ denote the incidence vector which captures all the edges incident on node u, *i.e.,* $\mathbf{I}_{uv}=1$ if an edge exists between nodes u and v, and otherwise 0. Finally, we introduce $\mathbf{b}_u$ to capture all the information associated with node u, *i.e.,* $\mathbf{b}_u=[\mathbf{x}_u; \mathbf{I}_u]$ denotes the concatenation of node attribute and incidence vector corresponding to node u. We also generate an augmented graph $\mathcal{G}'=(\mathcal{V}, \mathcal{E}', \tilde{\mathbf{X}})$, *i.e.,* for each node we generate a corresponding node in the augmented graph by slightly perturbing the attribute values, incident edges, and/or modifying u's sensitive attribute. For a GNN with K layers, the representations for node u for each layer is denoted as $\mathbf{h}_u^1, \cdots, \mathbf{h}_u^K$, where $\mathbf{z}_u=\mathbf{h}_u^K$ is representation at the last GNN layer. Analogously, $\tilde{\mathbf{z}}_u$ denotes the output representation of node u in $\mathcal{G}'$. The (dis)similarity between two node representations is given by a distance metric $D: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$. Our goal is to learn an encoder ENC which maps node u to its representation $\mathbf{z}_u$, *i.e.,* $\textsc{Enc}(u)=\mathbf{z}_u$. Lastly, a classifier f maps the representation $\mathbf{z}_u$ to a class label $\hat{y}_u$.

Graph Neural Networks. GNNs can be formulated as message passing networks (Wu et al., 2020) specified by trainable operators MSG, AGG, and UPD. In a K-layer GNN, the operators are recursively applied on $\mathcal{G}$, specifying how messages are exchanged between nodes, aggregated, and transformed to generate final node representations. A message between a pair of nodes (u, v) in layer k is defined as a function of hidden node representations from the previous layer as: $\mathbf{m}_{uv}^k=\textsc{Msg}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1})$. In AGG, messages from $\mathcal{N}_u$ are aggregated as: $\mathbf{m}_u^k=\textsc{Agg}(\mathbf{m}_{uv}^k | u \in \mathcal{N}_u)$. In UPD, the aggregated message $\mathbf{m}_u^k$ is combined with $\mathbf{h}_u^{k-1}$ to produce u's representation as: $\mathbf{h}_u^k=\textsc{Upd}(\mathbf{m}_u^k, \mathbf{h}_u^{k-1})$.

Fairness and Stability. Our goal is to learn fair and stable node representations. More specifically, the notions of fairness and stability that we consider in this work are counterfactual fairness and Lipschitz continuity, respectively.

Counterfactual Fairness: For graph representation learning, counterfactual fairness can be interpreted as node representations output by encoders should be independent of the sensitive attribute, *i.e.,* changing node u's sensitive attribute value should not affect the node representations.

Definition 1. *An encoder function* ENC *satisfies counterfactual fairness if the following holds for any given node u:*

$$\textsc{Enc}(u) = \textsc{Enc}(\tilde{u}^s), \tag{1}$$

where $\tilde{u}^s$ is a node in the augmented graph generated by modifying u's sensitive attribute values while keeping everything else constant.

Stability via Lipschitz Continuity: A function is stable according to Lipschitz continuity if slightly perturbing any given instance does not drastically change the output. In graph representation learning, this notion entails small perturbations to node attributes and/or incident edges should not drastically change the resulting representations.

Definition 2. *An encoder function* ENC *is stable according to the notion of Lipschitz continuity if:*

$$||\textsc{Enc}(\tilde{u}) - \textsc{Enc}(u)||_p \leq L||\tilde{\mathbf{b}}_u - \mathbf{b}_u||_p, \tag{2}$$

where $\tilde{u}$ is a node in the augmented graph generated by perturbing u's attribute values and/or incident edges, $\mathbf{b}_u$ and $\tilde{\mathbf{b}}_u$ capture the attribute and incident edge information for nodes u and $\tilde{u}$ respectively, and L is the Lipschitz constant.

3. Our Framework NIFTY

Next, we describe our framework NIFTY which aims to generate fair and stable graph embeddings by enforcing fairness and stability in the objective function as well as in the architecture of the underlying GNN.

Problem formulation (Fair and Stable embeddings). *Given* $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$, NIFTY *aims to generate embeddings* $\mathbf{z}_u \in \mathbb{R}^d$ *that are counterfactually fair (Eq. 1) and stable to attribute and structural perturbations of* $\mathcal{G}$ *(Eq. 2).*

Fairness and Stability in the Objective Function. To infuse fairness and stability in the objective function, we introduce a triplet-based objective that maximizes the agreement between the original graph and its counterfactual and noisy views. To this end, we use Siamese networks to maximize this agreement, *i.e.,* the augmented network neighborhoods and attribute vectors of the same node should result in the same embedding (Chen & He, 2020). Generating augmented views of graph structure and attribute information is key for the Siamese learning. We generate them using node-, sensitive attribute-, and edge-level perturbations. Refer Appendix D for details. To learn embeddings that are invariant to the sensitive attribute and stable against perturbations, we train the GNN encoder ENC using the Siamese framework (Bromley et al., 1994) and generate representations $\tilde{\mathbf{z}}_u$ of the augmented graph at every iteration. By generating augmented graphs, NIFTY can induce appropriate bias into the underlying GNN to learn embeddings invariant to the combination of counterfactual and random perturbations. A predictor $t: \mathbb{R}^d \rightarrow \mathbb{R}^d$ consisting of a fully-connected layer is then used to transform and match the representations with each other. Inspired by Grill et al. (2020), we define a triplet-based objective that optimizes the similarity between the graph and its augmented (*i.e.,* counterfactual and noisy) representations:

$$\mathcal{L}_s = \mathbb{E}_u\Big[\frac{1}{2}\big(D(t(\mathbf{z}_u), \text{sg}(\tilde{\mathbf{z}}_u)) + D(t(\tilde{\mathbf{z}}_u), \text{sg}(\mathbf{z}_u))\big)\Big], \quad (3)$$

where $t(\mathbf{z}_u)$ and $t(\tilde{\mathbf{z}}_u)$ are the transformed representations of u and $\tilde{u}$, D is the cosine distance, and stopgrad (sg) prevents gradients from being backpropagated. The sg signifies that the node representation $\tilde{\mathbf{z}}_u$ is considered as constant when operating on $t(\mathbf{z}_u)$ and vice-versa. The overall objective function for NIFTY is: $\min_{\theta_{\text{Enc}}, \theta_t, \theta_f} \mathbb{E}_u \left[(1-\lambda)\mathcal{L}_c\right] + \lambda\mathcal{L}_s$, where $\{\theta_{\text{Enc}}, \theta_t, \theta_f\}$ denotes trainable parameters of ENC, t, and classifier f, $\mathcal{L}_c$ is the binary cross entropy loss, and the expectation is taken over training nodes in $\mathcal{G}$. The regularization coefficient λ controls the trade-off between node classification loss $\mathcal{L}_c$ and the tripled-based objective $\mathcal{L}_s$. Algorithm 1 givens an overview of NIFTY algorithm.

Fairness and Stability in the GNN Architecture. NIFTY modifies the GNN's routing of neural messages. Typically, a GNN layer is given by (Sec. 2): $\mathbf{h}_u^k = \text{UPD}(\text{AGG}(\text{MSG}(\mathbf{h}_u^{k-1}, \mathbf{h}_v^{k-1}) | v \in \mathcal{N}_u), \mathbf{h}_u^{k-1})$. Considering AGG a fully-connected neural layer and UPD a non-linear activation function σ, we rewrite the message-passing step as: $\mathbf{h}_u^k = \sigma\big(\mathbf{W}_a^k\ \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1}\big)$, where $\mathbf{W}_n^k$ is the weight matrix associated with the neighbors of node u and $\mathbf{W}_a^k$ is the self-attention weight matrix at layer k. Definition 2 entails that the Lipschitz constant L provides an upper bound on how much u's node embedding can change. In fact, L represents the smallest value for which Eqn. 2 in Def. 2 holds true. Leveraging this understanding, NIFTY bounds the change in u's embedding by normalizing the encoder's weight matrices. This is because of the slope-restricted structure of the nonlinear activation function in the UPD step (proof in Sec. 4). Using our derivations in Sec. 4, we calculate the Lipschitz constant L of term $\mathbf{W}_a^k \mathbf{h}_u^{k-1}$ as the spectral norm σ of the weight matrix at each layer k. We use L to normalize $\mathbf{W}_a^k$ as: $\tilde{\mathbf{W}}_a^k = \mathbf{W}_a^k / \sigma(\mathbf{W}_a^k)$. We use this normalized weight matrix $\tilde{\mathbf{W}}_a^k$ to modify the UPD step: $\mathbf{h}_u^k = \sigma(\tilde{\mathbf{W}}_a^k\ \mathbf{h}_u^{k-1} + \mathbf{W}_n^k \sum_{v \in \mathcal{N}(u)} \mathbf{h}_v^{k-1})$. Lipschitz normalization has two advantages: 1) it bounds the difference between embeddings of original and perturbed node attributes; 2) it establishes a connection between stability and counterfactual fairness such that similar inputs should yield similar representations.

4. Theoretical Analysis of NIFTY

Next, we 1) prove that NIFTY's embeddings are stable, 2) provide a theoretical upper bound on unfairness of embeddings, and 3) show that downstream classifiers trained on NIFTY's embeddings are counterfactual fair.

Theorem 1 (NIFTY Stability). *Given a non-linear activation function σ that is Lipschitz continuous, the representations learned by our framework* NIFTY *are stable i.e.,*

$$||\text{ENC}(\tilde{u}) - \text{ENC}(u)||_p \leq \prod_{k=1}^{K} ||\mathbf{W}_a^k||_p ||(\tilde{\mathbf{b}}_u - \mathbf{b}_u)||_p, \quad (4)$$

where $\tilde{u}$ is a node in the augmented graph generated by perturbing u's attribute values and/or incident edges, $\mathbf{b}_u$ and $\tilde{\mathbf{b}}_u$ capture all attribute and edge information of nodes u and $\tilde{u}$, and $\mathbf{W}_a^k$ is u's self-attention weight at layer k.

Proof is provided in the Appendix A.

Theorem 2 (NIFTY Counterfactual Fairness). *Given a non-linear activation function σ that is Lipschitz continuous and a binary sensitive attribute s, the (counterfactual) unfairness of* NIFTY*'s representations can be bounded as:*

$$||\text{ENC}(\tilde{u}^s) - \text{ENC}(u)||_p \leq \prod_{k=1}^{K} ||\mathbf{W}_a^k||_p \quad (5)$$

where $\tilde{u}^s$ is a node in the augmented graph which is generated by modifying (flipping) the value of the sensitive attribute (s) of node u while keeping everything else constant.

Proof is provided in Appendix B.

Proposition 1 (Counterfactual Fairness of Downstream Classifier). *If* NIFTY*'s representations satisfy counterfactual fairness, then a downstream classifier $f: \mathbf{z}_u \rightarrow \hat{y}_u$ using those representations also satisfies counterfactual fairness.*

Proof is provided in the Appendix C.

5. Experiments

Next, we present experimental results for NIFTY framework. We first describe datasets designed to study fair and stable network embeddings and then outline experimental setup.

Datasets. We construct three new datasets. 1) The *German credit graph* has 1,000 nodes representing clients in a German bank connected based on the similarity of their credit accounts. The task is to classify clients into good vs. bad credit risks considering clients' gender as the sensitive attribute (Dua & Graff, 2017). 2) The *Recidivism graph* has 18,876 nodes representing defendants who got released on bail at the U.S. state courts between 1990-2009 (Jordan & Freiburger, 2015). Defendants are connected based on the similarity of past criminal records and demographics, where the goal is to classify defendants into bail (*i.e.,* unlikely to commit a crime if released) vs. no bail (*i.e.,* likely to commit a crime) considering race information as the protected attribute. 3) The *Credit defaulter graph* has 30,000 nodes representing individuals connected based on the similarity of their spending and payment patterns (Yeh & Lien, 2009).

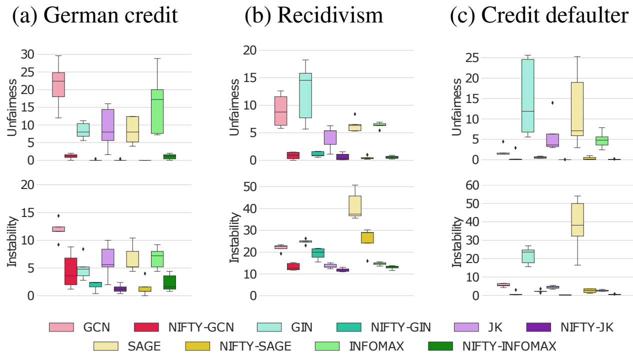

Figure 2. Unfairness and instability error rates for five GNNs and their NIFTY counterparts. NIFTY-enhanced GNNs give fairer and more stable predictions than their unmodified counterparts.

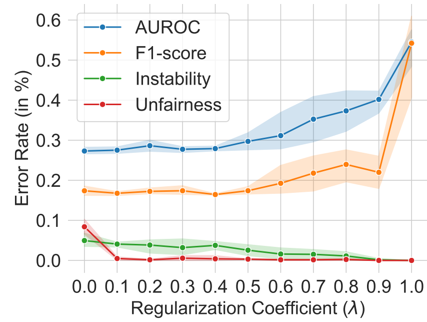

Figure 3. The effects of regularization on the performance of NIFTY-GIN for the German credit graph (see Fig. 4 for other datasets) where NIFTY achieves a near-perfect stability and fairness on the downstream task over a wide range of regularization.

The task is to predict whether an individual will default on the payment or not while considering age as the sensitive attribute. See Appendix E for details.

Metrics. We use AUROC and F1-score to measure GNN's predictive performance. To quantify group fairness, we use statistical parity (Δ_{SP}), and equal opportunity (Δ_{EO}) as defined in Dai & Wang (2021). To measure counterfactual fairness, we define the unfairness score as the percentage of test nodes for which predicted label changes when the node's sensitive attribute is flipped. Finally, the instability score represents the percentage of test nodes for which predicted label changes when random noise is added to node attributes.

GNN methods. We incorporate NIFTY into five GNN methods: GCN (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), Jumping Knowledge (JK) (Xu et al., 2018), GIN (Xu et al., 2019), and InfoMax (Veličković et al., 2019). Additionally, we consider two baseline methods: FairGCN (Dai & Wang, 2021) and RobustGCN (Zhu et al., 2019); all hyperparameters are set following the authors' guidelines. We use stop-gradient operation for training the Siamese networks (Chen & He, 2020). We set regularization coefficient to λ=0.6 and conduct a sensitivity analysis into the effect of λ on NIFTY's performance. See Appendix F for details.

Results: NIFTY improves fairness and stability. Across three datasets and five GNNs, NIFTY-modified GNNs learn fairer and more stable embeddings than unmodified GNNs (Fig. 2). On average, NIFTY improves stability and fairness of GNNs by 60.87% and 92.01%, respectively. Further, NIFTY can promote fairness and stability of GNNs without sacrificing their predictive performance, as evident by AUROC and F1-scores in Table 2. Finally, NIFTY outperforms baseline FairGCN and RobustGCN methods by 62.07% and 57.26% on four fairness and stability metrics (Table 1).

Results: NIFTY achieves group fairness. While NIFTY aims to capture counterfactual fairness, it remarkably improves group fairness of GNNs as it reduces information on protected attributes and makes the multi-objective problem of satisfying fairness and stability more tractable. Across three datasets, five GNNs, and two group fairness metrics, NIFTY achieves 43.56% lower Δ_{SP} and 34.70% lower Δ_{EO}, suggesting that in NIFTY, a node's chance of being represented as a particular point in the embedding space does not depend on its membership in a protected group.

Results: Fairness vs. stability vs. accuracy. As we increase regularization coefficient λ in NIFTY (Fig. 3), we find that the error rates for counterfactual fairness and stability steadily decrease. Even with a modest amount of regularization (λ=0.1), NIFTY achieves a 94.29% improvement in unfairness. As expected, a strongly regularized NIFTY model takes a hit on its predictive performance.

Results: Ablation study. We conduct ablations on two key NIFTY's components, namely the objective function and the Lipschitz layer-wise normalization. Results show that both components are necessary to generate fair and stable embeddings (Table 3). In particular, we observe a 90.7% improvement in fairness of NIFTY-GCN as compared to vanilla GCN, providing empirical evidence for our theoretical analysis that the Lipschitz normalization can improve both fairness and stability of graph embeddings (Sec. 4).

6. Conclusions & Future Work

We propose NIFTY, a unified framework that exploits a novel connection between counterfactual fairness and stability to learn network representations that are both fair and stable. NIFTY uses a two-level strategy to modify an existing GNN at the architectural and the objective function level. Results on new graph datasets from criminal justice and financial lending domains show that NIFTY improves fairness (counterfactual and group fairness) and stability without sacrificing predictive performance. This work paves way for several future directions, *e.g.,* extending NIFTY to generate fair and stable representations of other graph components (*e.g.,* edges, subgraphs) and to cater to other downstream tasks (*e.g.,* link prediction, graph classification).

References

Bose, A. J. and Hamilton, W. L. Compositional fairness constraints for graph embeddings. In *ICML*, 2019.

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a" siamese" time delay neural network. In *NeurIPS*, 1994.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

Chen, X. and He, K. Exploring simple siamese representation learning. *arXiv*, 2020.

Dai, E. and Wang, S. Fairgnn: Eliminating the discrimination in graph neural networks with limited sensitive attribute information. In *WSDM*, 2021.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Gouk, H., Frank, E., Pfahringer, B., and Cree, M. J. Regularisation of neural networks by enforcing lipschitz continuity. In *Machine Learning*. Springer, 2021.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

Huang, K., Xiao, C., Glass, L. M., Zitnik, M., and Sun, J. Skipgnn: predicting molecular interactions with skip-graph networks. In *Scientific Reports*, 2020.

Jin, G., Wang, Q., Zhu, C., Feng, Y., Huang, J., and Zhou, J. Addressing crime situation forecasting task with temporal graph convolutional neural network approach. In *ICMTMA*, 2020.

Jordan, K. L. and Freiburger, T. L. The effect of race/ethnicity on sentencing: Examining sentence type, jail length, and prison length. In *Journal of Ethnicity in Criminal Justice*. Taylor & Francis, 2015.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Liao, J., Huang, C., Kairouz, P., and Sankar, L. Learning generative adversarial representations (gap) under fairness and censoring constraints. *arXiv*, 2019.

Rahman, T. A., Surma, B., Backes, M., and Zhang, Y. Fairwalk: Towards fair graph embedding. In *IJCAI*, 2019.

Ustun, B., Spangher, A., and Liu, Y. Actionable recourse in linear classification. In *FAT*, 2019.

Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. Deep graph infomax. In *ICLR*, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. In *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K.-i., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *ICML*, 2018.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019.

Yeh, I.-C. and Lien, C.-h. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. In *Expert Systems with Applications*, 2009.

Zhu, D., Zhang, Z., Cui, P., and Zhu, W. Robust graph convolutional networks against adversarial attacks. In *KDD*, 2019.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. In *Bioinformatics*, 2018.