A Debiasing Framework for Graph Neural Networks Using Contrastive Learning

Kushal Bhatt
bkushal001@gmail.com

***Abstract*—Graphs are a prevalent data structure used across various domains, but they often inherit human biases that can propagate into downstream systems. For instance, recommendation systems for social networks may amplify gender or ethnic homophily, leading to biased recommendations that influence critical decisions. The increasing complexity of graph neural networks (GNNs) has made identifying and mitigating such biases particularly challenging. This paper introduces a novel debiasing framework for GNNs, leveraging the widely adopted contrastive learning paradigm. In contrastive learning, GNNs are trained to differentiate between observed and randomly generated graphs, with bias often serving as a useful feature for this discrimination task. Our approach counteracts this by explicitly introducing bias into the random graphs, rendering it an ineffective feature for discrimination and preventing its propagation into the model. By making the bias model explicit, our framework enhances control and transparency in bias mitigation. Evaluations on link prediction tasks demonstrate that our framework significantly reduces bias while preserving overall predictive accuracy, advancing the fairness and reliability of GNN-based systems.**

I. Introduction

The increasing reliance on algorithms in text and image processing has brought attention to human biases [1] and a critical need for fairness in the use of machine learning models [2], [3], [4], [5], [6]. Yet, another common and ubiquitous type of data—graphs—has received far less attention than they deserve. Graph data is not only the foundation of social and biological data [7] but ubiquitous across a huge range of computational problems [8], [9], [10]. For example, because social networks exhibit a strong homophily—the tendency for individuals to form connections with others who share similar characteristics such as gender, ethnicity, and culture [11]—AI systems trained on these social networks may make biased recommendations based on the homophily in high-stakes areas, such as professional networking that can influence hiring and promotions. For instance, it is well-known that the leadership roles in firms are dominated by white male population. *If* the recommendation engine of a professional networking service (e.g., LinkedIn) is optimized to predict the existence of actual professional acquaintance, the system is more likely to, from the homophily signal that it has learned from the existing data, recommend connections between white male professionals in the majority group, depriving opportunities from other populations and strengthening existing biases in the industry.

Let us demonstrate a potentially harmful consequence of network biases by using a small network of political books [12]. This network consists of 105 books written around the time of 2004 US presidential election consisting of 441 edges (co-purchase relation) between books. Many AI systems hinge on graph embedding [13], [14], [15], [16], where each node is represented as a point in space, with geometry reflecting the network structure. The network has a strong political discrimination, and so does the embedding generated by a standard graph embedding—LINE [17] (Fig.**??**B). Recommendation systems based on the embedding may promote books with a specific ideology and reinforce echo chambers while sidelining other viewpoints, ultimately exacerbating political polarization.

A common debiasing approach is based on the manipulation of the input data and output embedding, e.g., balancing the training data [18], [19], [20] or removing embedding dimensions associated with bias [2], [21]. However, these manipulation approaches may degrade embedding quality, fail to remove the bias, or even worse, introduce a new bias [21], [22], [23]. An alternative approach is adversarial learning, in which an adversarial model attempts to extract any biased features from the embedding, while the embedding is modified to be resistant to the adversary's attempts. However, the adversarial learning is often hindered by its high instability and sensitivity to hyperparameters [24].

We propose a simple yet effective framework for debiasing graph data that utilizes the inherent capacity of neural networks [25]. Our key idea is to use a *biased contrastive learning*, which trains a neural network to discriminate between actual data and random data. By introducing a specific bias into the random data, we ensure that sensitive attributes are not informative in the discrimination, *preventing biases from entering the neural network* (Fig. **??**C). The proposed training framework can be applied to a wide range of models from simple graph embedding methods (e.g., DeepWalk and node2vec) to deep graph neural networks (e.g., GCN and GAT), offering a flexible and practical alternative to existing debiasing methods. We test our approach by using link prediction task, demonstrating that our proposed framework substantially reduces biases while maintaining high link prediction accuracy as compared to the prevailing debiasing methods.

II. Method

A. Networks

We assume that a network consists of N nodes and M edges. Each node i has a C-dimensional vector x_i representing the i's attributes. Each edge (i, j) between nodes i and j can be directed and have weight w_{ij}. We assume that the given

network is weakly connected—every node is reachable from every other node when we ignore the direction of the edges.

B. Link prediction

Missing link prediction is a central task of graph representation learning and the basis of recommendation systems for social networks. We test graph neural networks with a standard benchmark of link prediction as follows. First, we randomly partition the set of edges in a given undirected network into a training set and a test set of almost an equal size while keeping the network constructed from the train edges being connected [13], [26]. We ensure the connectedness of the test network by adding the edges in a minimum spanning tree to the train set [13], [26]. Second, we generate negative edges—the pairs of unconnected nodes (i, j)—from the given network by sampling i and j uniformly at random. We generate the same number of negative examples as the number of test edges. Third, we train a graph embedding model using the network constructed with the train edge set. Fourth, for each test edge and negative edge, we calculate the likelihood of edge between two nodes by the dot similarity of the nodes' embedding vectors. We evaluate the likelihood of edge by using the area under curve of the receiver operating characteristics curve (AUC-ROC). We run the experiment five times with different random seeds.

C. Key idea

Graph neural network for link prediction is trained to differentiate the presence and absence of edges, a training framework called *contrastive learning* [25], [27], [28]. Contrastive learning has been used not only for link prediction but also for a variety of tasks including natural language processing and image recognition. For example, in a face recognition task, a neural network learns a person's face x by contrasting it with a reference face x' of a randomly-sampled person. The neural network recognizes target face x by focusing on the *differences* to the reference face x'. Now, if we sample reference x' from the same ethnic group as target x, the neural network learns the differences within the group, rather than learning ethnic traits. This can prevent the neural network from developing ethnic bias.

In context of graph embedding, a positive example is an edge $x = (i, j)$ between nodes i and j. A negative example (or negative edge) is a pair of unconnected nodes $x' = (i', j')$ in a given network [25]. As in the case of face recognition task, a graph embedding model learns the *difference* between the positive edge x and negative edge x'. Our key idea is to generate the negative edges with the same biases so that the positive and negative edges are indistinguishable in terms of the biases, which prevents the graph embedding model from learning the biases.

Let us demonstrate our key idea by using the political book network. Consider the political orientations as a sensitive attribute, and we want remove them from the embedding. The political attribute correlates with the network structure, namely a book is likely to be connected with the one with the same political leaning, resulting in many positive edges formed by books with the same political leaning (Fig. **??**E). Now, many graph neural networks including LINE are trained on negative edges that are sampled uniformly at random. Since the sampling of negative edges are independent of political leaning, the sampled negative edges do not have strong political homophily. This results in the misalignment of the positive and negative edges in terms of political homophily, which is then picked up by a graph embedding model as a useful feature for discrimination.

We prevent the model from using the sensitive attribute by generating negative examples with the same bias characteristics as the positive examples. To this end, we randomize the structure of a given network while preserving the joint probability of political leaning based on the maximum-entropy framework (Fig. **??**F; see the Model of network bias section). This results in a random network that is indistinguishable from the given network in terms of the political assortativity (Fig. **??**E), and the resulting graph embedding has no visible political separation accordingly (Fig. **??**C).

D. Contrastive learning for debiasing

We focus on graph neural networks that generate a node embedding based on the network structure and node features. We express a graph neural network as a function ϕ parameterized by $\vec{\theta}$ that produces an embedding of node i, i.e., $\phi(i; \vec{\theta}) \in R^{K\times 1}$. In link prediction, a graph neural network learns the probability $P(i, j)$ that an edge appears between two nodes (i, j), a common form being the softmax function, i.e.,

$$P(i, j) = \frac{1}{Z} \exp(\phi(i)^\top \phi(j)). \tag{1}$$

Variable Z is a normalization constant. Fitting the softmax function is notoriously difficult since Z is computationally demanding as it extends over all node pairs. Thus, several efficient estimation methods have been developed.

Noise contrastive learning (NCE) is an efficient estimation method for Eq. (1) [28]. The key idea underlying NCE is to train a different computationally-cheaper model whose best parameter θ^* in terms of the likelihood being the same as Eq. (1). Operationally, NCE trains a logistic regression model by classifying a set of node pairs (i, j) into *connected* node pairs and randomly-sampled node pairs, i.e.,

$$P\left(Y_{ij} = 1\right) := \frac{1}{1 + \exp(-f(i, j) + \ln P_0(i, j))}, \tag{2}$$

where $Y_{ij} = 1$ if nodes i and j are connected, $Y_{ij} = 0$ if they are not, and $f(i, j) = \phi(i)^\top \phi(j)$ is the node similarity. It is shown that NCE is asymptotically unbiased for an exponential probability model given by [28], [27]:

$$P(i, j) = \frac{1}{Z} \exp(f(i, j)), \tag{3}$$

which corresponds to the edge probability learned by a graph neural network given inEq. (1).

A key feature of NCE is that it is unbiased estimator agnostic to the choice of $P_0(i, j)$ because the effect of $P_0(i, j)$

is offset by $\ln P_0(i,j)$ in Eq. (2). Removing the offset gives rise to a powerful capacity of debiasing, as is demonstratd by [25]. Let us consider NCE without the offset $\ln P_0(i,j)$, i.e., discriminating the connected and unconnected node pairs by

$$P\left(Y_{ij}=1\right) := \frac{1}{1+\exp(-\phi(i)^\top \phi(j))} \tag{4}$$

Because we drop $\ln P_0(i,j)$, Eq. (4) gives a biased estimate. To see the effec of the bias, we derive a model for which Eq. 4 is unbiased for by rewriting Eq. (4) in form of NCE as

$$P\left(Y_{ij}=1\right) := \frac{1}{1+\exp\left(-f'(i,j)+\ln P_0(i,j)\right]} \tag{5}$$

where $f(i,j) := \phi(i)^\top \phi(j) + \ln P_0(i,j)$. Since Eq. (2) is unbiased estimator for Eq. (3), Eq. (5) is asymptomatically unbiased for

$$\begin{aligned} P(i,j) &= \frac{1}{Z'}\exp(f'(i,j)) \\ &= \frac{1}{Z'}P_0(i,j)\exp(\phi(i)^\top \phi(j)). \end{aligned} \tag{6}$$

Equation (6) decomposes the probability of edge $P(i,j)$ into $P_0(i,j)$ and node similarity $\phi(i)^\top \phi(j)$. From a different perspective, noise distribution $P_0(i,j)$ serves as a *null model* for networks, accounting for trivial relationships of nodes. The node similarity captures the *residuals* from $P_0(i,j)$, reflecting the non-trivial relationships not explained by the null model. This insight enables us to debias graph embedding, i.e., by introducing a bias into the null model, we can in turn remove the bias in the residual, from which the embedding is constructed.

The advantage of our approach is that the bias model is clear and explicit. All debiasing methods assume some null models because they must make judgment about when two nodes should be closer to each other than others. However, the assumptions are often unclear, even when the algorithm itself is simple. By making clear the bias model, we gain greater control over the process of debiasing and a deeper understanding of its consequences.

E. Models of network bias

The bias model should exhibit the same type and extent of bias while having no other structure. We construct such a bias model based on the maximum entropy principle [29], [30]. Specifically, we want the noise distribution $P_0(i,j)$ to be maximally random, i.e., maximizing the entropy

$$-\sum_{i,j} P_0(i,j)\log P_0(i,j), \tag{7}$$

while preserving the assortativity over the protected groups:

$$\sum_{i\in\mathcal{C}_\ell}\sum_{i\in\mathcal{C}_k} P_0(i,j) = \sum_{i\in\mathcal{C}_\ell}\sum_{i\in\mathcal{C}_k} P(i,j), \tag{8}$$

for all $1 \le k,\ell \le L$, where L is the number of unique protected groups, and $\mathcal{C}_k$ is the set of nodes in the kth group. Finding the $P_0(i,j)$ is a constrained optimization problem, which can be formulated as a Lagrangian:

$$\begin{aligned} \mathcal{L} := &-\sum_{i,j} P_0(i,j)\log P_0(i,j) \\ &+\sum_{k=1}^{L}\sum_{\ell=k}^{L}\beta_{k\ell}\left(\sum_{i\in\mathcal{C}_\ell}\sum_{i\in\mathcal{C}_k} P_0(i,j) - \sum_{i\in\mathcal{C}_\ell}\sum_{i\in\mathcal{C}_k} P(i,j)\right). \end{aligned} \tag{9}$$

By taking a functional derivative with respect to P_0 and solving $\partial\mathcal{L}/\partial P_0 = 0$, we obtain

$$P_0(i,j) = \text{Poisson}(\lambda_{ij}), \tag{10}$$

$$\lambda_{ij} = \frac{1}{|\mathcal{C}_{c_i}||\mathcal{C}_{c_j}|}\sum_{i\in\mathcal{C}_\ell}\sum_{j\in\mathcal{C}_k} P(i,j). \tag{11}$$

This bias model is maximally random while preserving the same type and extent of bias in the given network. One can have additional constraints to model a more complex network bias, i.e., degree heterogeneity, degree-degree correlation, and edge directionality. See [29], [30] for other network models based on the maximum entropy principle.

III. Results

A. Datasets

We test GNNs with four different networks (TableI): the political books [**?**], the political blogs [31], the airport network [32], twitch gamers [33] and Facebook network [34]. These networks include categorical attributes for nodes that we do not want to use to make link prediction (i.e., protected attributes). For all networks, we ignore the edge directionality.

The political blog network represents the political blogs about the 2004 US presidential election connected by hyperlinks. We use the political affiliation assigned to the blogs as the protected attribute.

The airport network is the network of airports connected by edges indicating direct commercial flight. We use the geographical region of the airports as the protected attribute.

Twitch dataset is a social network of Twitch gamers which was collected using public Twitch API. Graph represents a single strongly connected component and edges are the mutual follower relationships between them. We have few features as the metadata for nodes, but for this paper we used language as the protected attribute.

Facebook dataset used here is a friendship network of 100 american colleges and universities. We use gender as a protected attribute. Nodes for which gender is unknown are given a separate group.

B. Graph neural network models

We test three graph neural networks with different architectures: word2vec (a neural network with one hidden linear layer and the softmax output layer) [35], GCN [36], and GAT [**?**]. The GCN and GNN needs node features as input, for which we use $K = 128$ dimensional embedding vectors generated by DeepWalk. We find that using node2vec instead of DeepWalk yield qualitatively the same results (Appendix). To see the

TABLE I: Statistics of network data for link prediction. Information modularity indicates the strength of the communities in a network, where each community is a set of nodes with the same protected attribute. See **??**add the robustness modularity paper.

2*Network	2*Nodes	2*Edges	Protected category (# of categories)	2*Information Modularity
Political Books	105	441	Political leaning (3)	0.061
Political Blogs	1224	16715	Political leaning (3)	0.0925
Airport	2898	15564	Geographical regions (5)	0.112
Twitch	168114	6797557	Language (20)	0.05
Facebook	41554	1362229	Gender (3)	-0.004

effect of debiasing, we compare each model trained with the proposed bias sampling of negative examples with the one with with a conventional uniform sampling. We also employ two families of debiasing methods as baselines. One family is based on the manipulating network data to balance the training data for DeepWalk, i.e., FairWalk [19] and CrossWalk [18]. The other family is based on the manipulation of the output embedding, for which we use a debiasing method based on linear projection [2]. For a stable baseline we use the method proposed in [2]. To apply this method to a graph, we use PCA to determine the bias direction using 20% of the nodes nearest to group centroids. As opposed to the previous work we do not have any gender neutral nodes. We use all the nodes as equalize nodes. We change embedding of each node as

$$\overrightarrow{W} := \overrightarrow{W} - \overrightarrow{D} \cdot proj_{\overrightarrow{W}} \overrightarrow{D} \tag{12}$$

where $\overrightarrow{D}$ is the bias direction.

C. Bias in graph embedding

We quantify the biases in graph embedding by the extent to which the sensitive attribute shapes the structure of the embedding. If we were to have an well-debiased graph embedding, the sensitive attributes should not be the main dimensions of the graph embedding. Following the previous study [2], we identify the embedding dimensions that are the most coherent to sensitive attributes and then quantify the ratio of explained variance to total variance of the nodes in the embedding as the level of bias (Fig. **??**). We find that ...

While the sensitive attributes may not entirely shape the embedding structure, they could still have a local impact on the proximity of nodes. To examine the impact on the local proximity, we test to what extent each node is equidistance to the protected groups. More specifically, for each node i with embedding vector $\vec{u}_i$, we compute the cosine similarity (i.e., $\cos(\vec{u}_i, \vec{u}_\ell)$) to the centroid vector $\vec{u}_\ell$ of each group $c = \ell$. Then, we compute the *local disparity* D_i by the variance over all groups, i.e.,

$$D_i := \frac{1}{L} \sum_{\ell=1}^{L} \left(\cos(\vec{u}_i, \vec{u}_\ell) - \frac{1}{L} \sum_{\ell'=1}^{L} \cos(\vec{u}_i, \vec{u}_{\ell'}) \right)^2 . \tag{13}$$

We find that the local disparity drops for majority of nodes in the network when trained using biased negative sampling (Fig. **??**). The reduction of disparity is overall consistent across different model architecture and networks.

In summary, our training framework consistently brings a reduction of biases in graph neural networks compred to their unbiased counterparts across different networks both at global and local levels. The reduction of bias is comparable or substantial compared to the baseline debiasing methods. These results demonstrate that our training framework for debiasing is consistently effective and model-independent in reducing the biases in graph embedding.

D. Prediction performance

Debiasing an embedding involves masking certain information within the embedding, which can potentially lead to a significant reduction in the embedding's utility. Indeed, the debiasing methods including our method and the baselines all results in the decreased link prediction performance. However, the amount of the decrease differs method by methods. In fact, our training method results in the least decreases in the performance among all the debiasing methods (Fig. xxx).

We also examine the utility of embedding at a local node. For each node, we find the 50 closest neighbors in the given embedding. Then, we calculate the average precision of the node similarity for the test edges.

IV. Conclusion

We proposed a novel training framework to jointly improve fairness and quality of graph embeddings. We validated the proposed framework by training three models ranging from a shallow to deep neural networks. Notably, our approach makes the biased structural features inconseqential to model training, rather than heuristically removing biased input data or biased dimensions in the generated embedding. Overall, our proposed framework presents a new way to address the issue of biases in graph data, and can be applied to a variety of downstream AI applications.

There are certain limitations to be acknowledged. First, our approach requires a biased model encoding biased associations. We used the degree-corrected stochastic block model as the biased model by following the previous study [25]. Learning a suitable biased model for debiasing warrants future work. Second, our approach assumes that the sensitive attribute is known and labeled, which may not always be the case in real-world scenarios. Third, while we have focused on

direct associations between sensitive attributes, our approach may not completely eliminate bias due to indirect associations of biased attributes through other attributes [37]. Despite these limitations, our study demonstrates that by biasing the sampling of negative examples, it is possible to reduce bias while maintaining high accuracy in graph embedding.

REFERENCES

[1] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big? ," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 610–623. [Online]. Available: https://doi.org/10.1145/3442188.3445922

[2] T. Bolukbasi, K. Chang, J. Y. Zou, V. Saligrama, and A. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," *CoRR*, vol. abs/1607.06520, 2016. [Online]. Available: http://arxiv.org/abs/1607.06520

[3] S. Bourli and E. Pitoura, "Bias in knowledge graph embeddings," in *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 6–10, ISSN: 2473-991X.

[4] "Gender bias on wikipedia," page Version ID: 1128262996. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Gender_bias_on_Wikipedia&oldid=1128262996

[5] Google fixes translate tool after accusations of sexism. Section: Lifestyle. [Online]. Available: https://www.independent.co.uk/life-style/women/google-translate-sexist-masculine-feminine-he-said-she-said-english-spanish-language-a8672586.html

[6] T. S. Pages. Nikon camera says asians: People are always blinking - sociological images. [Online]. Available: https://thesocietypages.org/socimages/2009/05/29/nikon-camera-says-asians-are-always-blinking/

[7] M. Newman, *Networks*, 2nd ed. Oxford University Press.

[8] J. M. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Zídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. A. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, pp. 583–589, 2021.

[9] A. Derrow-Pinion, J. She, D. Wong, O. Lange, T. Hester, L. Perez, M. Nunkesser, S. Lee, X. Guo, B. Wiltshire, P. W. Battaglia, V. Gupta, A. Li, Z. Xu, A. Sanchez-Gonzalez, Y. Li, and P. Velickovic, "ETA prediction with graph neural networks in google maps," *CoRR*, vol. abs/2108.11482, 2021. [Online]. Available: https://arxiv.org/abs/2108.11482

[10] A. Mirhoseini, A. Goldie, M. Yazgan, J. W. Jiang, E. M. Songhori, S. Wang, Y.-J. Lee, E. Johnson, O. Pathak, A. Nazi, J. Pak, A. Tong, K. Srinivasa, W. Hang, E. Tuncer, Q. V. Le, J. Laudon, R. Ho, R. Carpenter, and J. Dean, "A graph placement methodology for fast chip design," *Nature*, vol. 594 7862, pp. 207–212, 2021.

[11] S. Yang, "Networks: An introduction by m. e. j. newman," *The Journal of Mathematical Sociology*, vol. 37, no. 4, pp. 250–251, 2013. [Online]. Available: https://doi.org/10.1080/0022250X.2012.744247

[12] J. Kunegis, "KONECT – The Koblenz Network Collection," in *Proc. Int. Conf. on World Wide Web Companion*, 2013, pp. 1343–1350. [Online]. Available: http://dl.acm.org/citation.cfm?id=2488173

[13] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. Association for Computing Machinery, Jul. 2016, pp. 855–864, arXiv:1607.00653 [cs, stat]. [Online]. Available: https://doi.org/10.1145/2939672.2939754

[14] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *CoRR*, vol. abs/1706.02216, 2017. [Online]. Available: http://arxiv.org/abs/1706.02216

[15] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '14. Association for Computing Machinery, pp. 701–710. [Online]. Available: https://doi.org/10.1145/2623330.2623732

[16] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017. [Online]. Available: https://arxiv.org/abs/1710.10903

[17] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. International World Wide Web Conferences Steering Committee, pp. 1067–1077. [Online]. Available: https://doi.org/10.1145/2736277.2741093

[18] A. Khajehnejad, M. Khajehnejad, M. Babaei, K. P. Gummadi, A. Weller, and B. Mirzasoleiman, "CrossWalk: Fairness-enhanced node representation learning." [Online]. Available: http://arxiv.org/abs/2105.02725

[19] T. Rahman, B. Surma, M. Backes, and Y. Zhang, "Fairwalk: Towards fair graph embedding," pp. 3289–3295. [Online]. Available: https://www.ijcai.org/proceedings/2019/456

[20] I. Solaiman and C. Dennison, "Process for adapting language models to society (PALMS) with values-targeted datasets," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., pp. 5861–5873. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/2e855f9489df0712b4bd8ea9e2848c5a-Abstract.html

[21] S. Ravfogel, Y. Elazar, H. Gonen, M. Twiton, and Y. Goldberg, "Null it out: Guarding protected attributes by iterative nullspace projection," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7237–7256. [Online]. Available: https://aclanthology.org/2020.acl-main.647

[22] A. Bose and W. Hamilton, "Compositional fairness constraints for graph embeddings," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, pp. 715–724, ISSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v97/bose19a.html

[23] H. Edwards and A. J. Storkey, "Censoring representations with an adversary," *CoRR*, vol. abs/1511.05897, 2015.

[24] Y. Xing, Q. Song, and G. Cheng, "On the algorithmic stability of adversarial training," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26 523–26 535, 2021.

[25] S. Kojaku, J. Yoon, I. Constantino, and Y.-Y. Ahn, "Residual2vec: Debiasing graph embedding with random graphs," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., pp. 24 150–24 163. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/ca9541826e97c4530b07dda2eba0e013-Abstract.html

[26] J. Yoon, K.-C. Yang, W.-S. Jung, and Y.-Y. Ahn, "Persona2vec: a flexible multi-role representations learning framework for graphs," *PeerJ Computer Science*, vol. 7, p. e439, Mar. 2021. [Online]. Available: https://doi.org/10.7717/peerj-cs.439

[27] C. Dyer, "Notes on noise contrastive estimation and negative sampling," 2014. [Online]. Available: https://arxiv.org/abs/1410.8251

[28] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 297–304, ISSN: 1938-7228. [Online]. Available: https://proceedings.mlr.press/v9/gutmann10a.html

[29] G. Cimini, T. Squartini, F. Saracco, D. Garlaschelli, A. Gabrielli, and G. Caldarelli, "The statistical physics of real-world networks," *Nature Reviews Physics*, vol. 1, no. 1, pp. 58–71, 2019.

[30] M. Bardoscia, P. Barucca, S. Battiston, F. Caccioli, G. Cimini, D. Garlaschelli, F. Saracco, T. Squartini, and G. Caldarelli, "The physics of financial networks," *Nature Reviews Physics*, vol. 3, no. 7, pp. 490–507, 2021.

[31] L. A. Adamic and N. Glance, " The Political Blogosphere and the 2004 U.S. Election: Divided They Blog ," in *Proceedings of the 3rd International Workshop on Link Discovery*, ser. LinkKDD '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 36–43. [Online]. Available: https://doi.org/10.1145/1134271.1134277

[32] "Openflights: Airport and airline data," https://openflights.org/data.html, (Accessed on 01/12/2023).

[33] B. Rozemberczki and R. Sarkar, "Twitch gamers: a dataset for evaluating proximity preserving and structural role-based node embeddings," 2021.

[34] A. L. Traud, P. J. Mucha, and M. A. Porter, "Social structure of facebook networks," *CoRR*, vol. abs/1102.2166, 2011. [Online]. Available: http://arxiv.org/abs/1102.2166

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, vol. 26. Curran Associates, Inc. [Online]. Available: https://proceedings.neurips.cc/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html

[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[37] H. Gonen and Y. Goldberg, "Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them," *CoRR*, vol. abs/1903.03862, 2019. [Online]. Available: http://arxiv.org/abs/1903.03862