RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] CrossWalk Fairness-enhanced Node Representation Learning

Gijs Joppe Moens[1], Job de Witte[1], Tobias Pieter Göbel[1], and Meggie van den Oever[1]
[1]University of Amsterdam, Amsterdam, The Netherlands

Edited by
Koustuv Sinha,
Maurits Bleeker,
Samarth Bhargav

Received
04 February 2023

Published
20 July 2023

DOI
10.5281/zenodo.8173747

Reproducibility Summary

"CrossWalk: Fairness-Enhanced Node Representation Learning" is set to be reproduced and reviewed. It presents an extension to existing graph algorithms that incorporate the idea of biased random walks for obtaining node embeddings. CrossWalk incorporates fairness by up-weighting edges of nodes located near group boundaries. The authors claim that their approach outperforms baseline algorithms, such as DeepWalk and FairWalk, in terms of reducing the disparity between different classes within a graph network.

The authors accompanied their paper with the publication of an open GitHub page, which includes the source code and relevant data sets. The limited size of the data sets in combination with the efficient algorithms enables the experiments to be conducted without significant difficulties and is computable on standard CPUs without the need for additional resources.

In this reproducibility report, the outcomes of the experiments are in agreement with the results presented in the original paper. However, the inherent randomness of the random walks makes it difficult to quantify the extent of similarity between the reproduced results and the results as stated in the original paper. However, it can be concluded that CrossWalk results in a decreased disparity between groups in graph networks.

The authors effectively conveyed the underlying concept of their proposed method, rendering it both intriguing and straightforward to comprehend the key ideas. Furthermore, the authors successfully incorporated a range of methods and baseline algorithms into the paper.

In contrast, the source code may not have been optimally constructed with reproducibility in mind. Certain sections of the code appear to be unfinished or inadequately executed. Additionally, the authors neglected to specify key hyperparameters, resulting in the unidentifiability of certain results. This presents challenges in drawing conclusions based on the available sources.

The authors were unable to respond in time for elaborating on certain implementation details. However, we did receive additional data which was crucial to obtaining certain results.

1 Introduction

The usage of machine learning has increased significantly in recent years. However, fairness in machine learning is only a recent research topic and has not much extended to advanced concepts, such as graphs. Graphs are widely applicable and have great potential in many fields. Hence, it is crucial that fairness is implemented in such algorithms. The paper 'CrossWalk: Fairness-Enhanced Node Representation Learning' that will be reproduced and reviewed introduces an extension upon existing graph algorithms that use the concept of random walks to obtain node embeddings [1].
Graph representation learning algorithms map a graph to a vector space while preserving its structural information. Some frequently used approaches are based on DeepWalk, a random-walk approach to graph representation learning. Fairness in these graphs can be enhanced by motivating the random walks to take steps that cross group boundaries. The Fairwalk method (Rahman et al.) [2] introduced this concept by increasing the weights of edges between nodes of distinct groups. Khajehnejad et al. take it a step further by additionally increasing the weights of edges that are close to group boundaries, thereby blurring group boundaries and improving fairness while preserving the structure. Originally, CrossWalk is tested only on graphs including no more than three groups. However, in many practical scenarios, graphs with a larger number of distinct groups may be encountered. In order to test CrossWalk's ability to generalize, we will test the performance of its embeddings on influence maximization on synthetic graphs containing more groups. Additionally, to improve CrossWalk's generalization to graphs containing more groups, a new measure of proximity is proposed. The core idea of this proposed measure is that for graphs with a larger number of groups, nodes that lie around several boundaries influence several groups and their edges should be weighted more heavily.

2 Scope of reproducibility

2.1 Original paper

To evaluate the effectiveness of CrossWalk, we compare its node embeddings with formerly introduced approaches DeepWalk and FairWalk. We test the following claim: fairness is enhanced while performance is preserved using node embeddings obtained by CrossWalk as opposed to benchmark methods DeepWalk and FairWalk considering the following metrics:*Influence maximization(1), Node classification(2) and Link prediction (3)*

2.2 Extension

To extend the existing research we explore the application of CrossWalk on graphs that are made up of more groups and consider a tuned version of the measure of proximity. To do so we introduce the following hypotheses:

- The superiority of CrossWalk, compared to FairWalk and DeepWalk, is maintained on graphs containing a larger number of groups. (4)
- We propose a novel approach for fairness-enhanced node learning MoonWalk improving upon the existing CrossWalk method. (5)

3 Methodology

3.1 Embedding Methods

Node2Vec — Node2Vec [3] is an algorithm that creates latent embeddings for a given graph. It will return an embedding $v \in \mathbb{R}^d$ for every node, with the objective of maintaining the internal structure of the original graph.
First, Node2Vec initiates random walks, by stepping stochastically through the graph. When considering a weighted graph, this happens according to the edge weights. Another way of guiding the random walk is by introducing a search bias α. Suppose the random walk just stepped from node t to node v, then the transition probability from v to x, becomes $\pi_{vx} = \alpha(t,x)w_{vx}$, with α defined as follows:

$$\alpha(t,x) = \begin{cases} \frac{1}{q} & \text{if } d_{tx} = 0 \\ 1 & \text{if } d_{tx} = 1 \\ \frac{1}{q} & \text{if } d_{tx} = 2 \end{cases} \tag{1}$$

Here, d_{tx} represents the shortest path distance between node t and node x. Intuitively, p gives a measure for how likely the random walk will return to the previous node (Breadt First Search, BFS), whereas q represents to what extent the random walk will explore nodes far away from the root node (Depth First Search, DFS)[3].
With the sequences of nodes, produced by the random walks, a Skip-Gram [4] model is trained. Here, the nodes are interpreted as words and the walks are interpreted as sentences. The Skip-Gram model will output the embeddings for each node.

DeepWalk — Deepwalk is a special case of the Node2Vec algorithm. More specifically, parameters p, q are set to 1, such that $\alpha = 1$. In this case, the chances of a transition in the random walk are purely decided on the edge weights.
For simplicity, we (and the original paper) mainly use DeepWalk for creating embeddings in the experiments shown.

3.2 Re-weigthing algorithms

The fairness of graph embeddings can be enhanced by modifying the existing weights. This will guide the random walks differently through the graph, and will therefore create different embeddings. The key idea is that edges that are closer to the group boundary are up-weighted, such that the random walks will cross groups more often. In this section, we will cover three re-weighting algorithms that attempt to enhance the fairness of the node representations.

FairWalk — FairWalk [5] was first introduced by Rahman et al. The FairWalk algorithm only re-weights inter-group connections. And does so, such that from a boundary node, each group is equally likely to be visited. This method is viewed as a benchmark method for re-weighing graph edges.

CrossWalk — CrossWalk [1] is claimed to be an improvement upon FairWalk by the authors of the original paper. CrossWalk not only reweighs edges in between different groups but changes weights based on the distance to a group boundary.
This distance to the boundary of node v, $m(v)$, is measured by the fraction of times r truncated random walks of length d, starting from root node v, visit a node u for which $l_v \neq l_u$. Where we define l_w to be the group that node w belongs to. Written more precisely as:

$$m(v) = \frac{\sum_j^r \sum_{u \in W_v^j, l_u \neq l_v} 1}{rd} \tag{2}$$

Here we used W_v^j to be the set of nodes visited by the j_{th} random walk from node v. When $m(v)$ is obtained for every node v in the graph, CrossWalk adjusts edge weights w_{uv} to $\tilde{w}_{uv}$ as follows:

$$\tilde{w}_{uv} = \begin{cases} w_{vu}(1-\alpha)\frac{m(u)^p}{\sum_{z \in N_v} w_{vz} m(z)^p} & \text{if } l_u = l_v \\ \\ w_{vu}\alpha\frac{m(u)^p}{|R_v| \sum_{z \in N_v^c} w_{vz} m(z)^p} & \text{if } l_v \neq l_u = c \end{cases} \tag{3}$$

Where we define $N_v = \{u \in \mathcal{N}(v) : l_u = l_v\}$ and $N_v^c = \{u \in \mathcal{N}(v) : l_v \neq l_u = c\}$, with $\mathcal{N}(v)$ being the set of neighbouring nodes of v. R_v is the set of nodes in the neighbourhood of v, that belong to a different class: $R_v = \{u \in \mathcal{N}(v) : l_v \neq l_u\}$. Here parameter α will influence the chance of crossing a group boundary in the random walk, while p parametrizes the importance of the distance to the boundary.

MoonWalk — In this paper, we propose MoonWalk, an extended version of CrossWalk. The difference with respect to CrossWalk is embedded in the calculation of the distance to the group boundary $m(v)$. Originally, this is measured by counting the number of times a random walk visits a node in another group (See (2)). However, we propose an alternative measure $\mu_P(v)$, that also captures the number of different groups a random walk falls upon.

Let $M^c(W_v)$ be the number of times a random walk W_v visits a node in group c, excluding the class of node v. More formally this becomes:

$$M^c(W_v) = |\{u \in W_v : l_v \neq l_u = c\}| \tag{4}$$

Now, for every iterated random walk we measure the following quantity:

$$\xi_P(W_v) = \frac{(\sum_c M^c(W_v))^2}{\sqrt[P]{\sum_c M^c(W_v)^P}} \tag{5}$$

Where P, a parameter that occurs in the denominator, as the P-norm of $M(W_v)$.
The expression for $\mu_P(v)$ will then be written as:

$$\mu_P(v) = \frac{1}{rd}\sum_j^r \xi_P(W_v^j) \tag{6}$$

With again r, the number of random walks, and d, length of each walk. The weights would then be updated according to (3), with $m(v)$ replaced by $\mu_P(v)$. Note that for $p = 1$ MoonWalk reduces to Crosswalk: $\mu_1(v) = m_(v)$. Also, if the graph consists of 2 groups only, MoonWalk will be equivalent to CrossWalk:$\mu_P(v) = m_(v) \quad \forall P$.

When considering 3 groups or more, MoonWalk will up-weight edges to nodes that are closer to multiple boundaries. CrossWalk, on the other hand, will not distinguish between one boundary or multiple boundaries.

Notice that, as opposed to $m(u)$, $\mu_P(u)$ is not necessarily bounded from above by 1. One might expect that this is problematic since we raise this value to a power p. We argue however, that the result will be equivalent with the use of a $\mu_P^{'}(u) = \frac{1}{C}\mu_P(u) \leq 1 \quad \forall u$, since any normalization factor $\frac{1}{C}$ would be factored out by the denominator of equation (3). This argument also shows the redundancy of the denominator rd in equation (2).

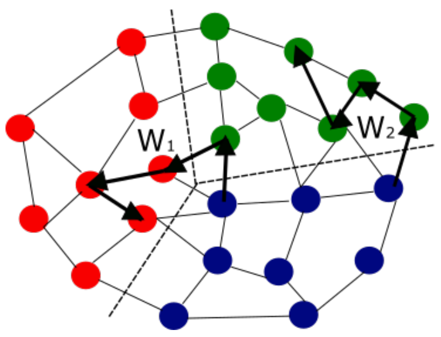


Figure 1. In this figure a graph with 3 groups (red, green, blue) is shown. W_1 and W_2 depict two different random walks of length 4. Both W_1 and W_2 visit 4 nodes that belong to other groups. However, for $p > 1$: $\xi_P(W_1) \neq \xi_p(W_1)$. For example: $\xi_2(W_2) = \frac{4^2}{\sqrt{4^2}} = 4$ and $\xi_2(W_1) = \frac{4^2}{\sqrt{1^2+3^2}} = \frac{16}{\sqrt{10}} > 4$. Cross-Walk, would weigh these walks equally well.

3.3 Metrics

The node representations, created by the methods described above, are evaluated on three different metrics. The fairness is measured by the disparity between the given groups and is defined below.

Influence Maximization — Starting at a certain node in the graph, the influence of that node could be measured as the infected individuals through an Independent Cascade model [6]. In order to get a measure that represents the influence of the whole graph well, the most influential nodes, are used as starting node(s). The algorithm used for for finding these points is k-medoids which randomly initiates k medoids and iteratively assigns nodes to clusters and updates the medoids accordingly. These most influential nodes serve as seeds for the IC model to obtain the number of infected individuals. In this setting, influence is measured as the percentage of foreign nodes reached [7].

Link Prediction — Node embeddings can be used to predict edges in a graph. This can be implemented as a logistic regression model on the edges where the feature vector of a certain edge between nodes v and u, with their node embeddings $\vec{v}$ and $\vec{u}$, is formulated as $(\vec{v} - \vec{u})^{\circ 2}$, denoting the Hadamard power with $\circ$. Performance of this task is measured as the accuracy of the model where it is evaluated on positive samples that have been excluded while training, and on an equal amount of generated negative samples.

Node Classification — Node embeddings can be used to identify their corresponding node through node classification where a model maps from the embedding space to the original space. This can be done semi-supervised with the machine learning algorithm Label Propagation, whose performance is measured by its accuracy.

Disparity — The fairness of a task can be measured with the metric disparity. Intuitively, an algorithm is said to be fair when its performance on the different groups of the graph is consistent. Let A be an algorithm and $Q \in \mathbb{R}$ and $Q_i \in \mathbb{R}$ be its overall performance and performance on group i respectively. Then A is said to be fair when Q_i is similar to Q_j for all the groups in the graph. This similarity will be calculated as the variance:

$$\text{disparity}(A) = Var(Q_i : i \in [C])$$

Decreasing the disparity for different metrics will make the model's decisions less responsive to a sensitive attribute, which can be seen as a form of demographic parity. [8]

3.4 Datasets

The utilization of the datasets will be briefly outlined below, with further details available in the original Crosswalk paper.

Rice-Facebook — The real-world Rice-Facebook dataset [9] includes Facebook friendship connections between students of Rice university. This graph is made up of two groups separated by age

Twitter — The real-world Twitter dataset is a subgraph of a large Twitter dataset [10] in which users are connected through tweets. The graph is made up of three groups based on the user's political standpoint.

Synthetic — Besides these real-world datasets, the original paper considers two synthetic datasets characterised by the number of classes $k \in \{2, 3\}$

Additional datasets — Furthermore, we test on synthetic datasets with more than three groups. The motivation for this extension of research is twofold. Firstly, increasing the number of clusters in the synthetic graphs covers a regime of real-life graphs that have not yet been explored in the published research. Real-life graphs are often subdivided into many small clusters rather than a few larger ones. A priori, it is not obvious whether the reported superiority of CrossWalk extends to this regime. No mention of the possible effects of increasing the number of groups is made in the original paper. Importantly, by merely increasing the number of groups we compromise on realism as well since the number of nodes per group decreases. To counter this effect, we accordingly increase the number of nodes with respect to the synthetic graphs used in the original paper.
Besides providing a new testing ground for CrossWalk, graphs with three or more groups allow us to compare it to our proposed MoonWalk. In general, we expect the effect of MoonWalk with respect to CrossWalk to increase when increasing the number of groups. As a minor difference with the original approach, we allow for some randomness by drawing the intra- and intergroup probabilities from uniform probability distributions $p^i_{INTRA} \sim \mathcal{U}(p_{hom,min}, p_{hom,max})$ and $p^{i,j}_{INTER} \sim \mathcal{U}(p_{het,min}, p_{het,max})$ for all groups i and j.

3.5 Hyperparameters

Hyperparameters for Node2Vec/Deepwalk — Throughout this research, we have constricted ourselves to use an embedding dimension of size $d = 128$, in correspondence with the original paper. Furthermore, we set the values of p, q, as defined in equation (1), to $p = 1, q = 1$ such that we work solely with DeepWalk unless specifically mentioned. We fix the random walk length for DeepWalk to a value of 40, and the number of walks to 30. The window size is set to equal 10.

Hyperparamaters for CrossWalk — For CrossWalk we have set the random walk length to $d = 5$, and the number of walks to $r = 500$. The parameters α and p are varied among our experiments and are explicitly mentioned. In the comparison to MoonWalk, we increase the random walk length to $d = 8$, in order to capture the difference.

Hyperparameters for MoonWalk — MoonWalk has the additional parameter P. In this research we have used the following values: $P \in \{1, 2, 3, 4\}$. Where $P = 1$ gives an equivalent method to CrossWalk and is therefore useful as a sanity check for interpreting the results. In the results, MoonWalk with $P = 3$ is shown.

Hyperparameters for Influence Maximization — Activation probability is set to 0.03 for synthetic data experiments and 0.01 for real-world data (Twitter, Facebook). We take the number of K-Medoids to be 40.

3.6 Implementation

Khajehnejad et al. published an open GitHub page along with their paper including source code and datasets. The GitHub repository can be found at https://github.com/ ahmadkhajehnejad/CrossWalk, which includes the following directories: **/data**, **/deepwalk**, **/influence_maximization**, **/link_prediction** and **/node_classification**. Inside the **/data** folder, the authors provided implementations for FairWalk and CrossWalk.
On top of that, a notebook is given, that shows figures of hard-coded data. In this project, we aim to re-calculate all embeddings using the code provided. All results given are based on the average performance on 5 independent runs. The experiments were run on a CPU, taking 10 minutes at most per setting.

4 Results

4.1 Results reproducing original paper

Result 1 — Figure 2 shows the performance of influence maximization on embeddings obtained by DeepWalk, FairWalk, and CrossWalk. The same trend is observed as in the original paper where CrossWalk experiences a larger decrease in disparity while the influence stays consistent, and thereby supports statement 1.

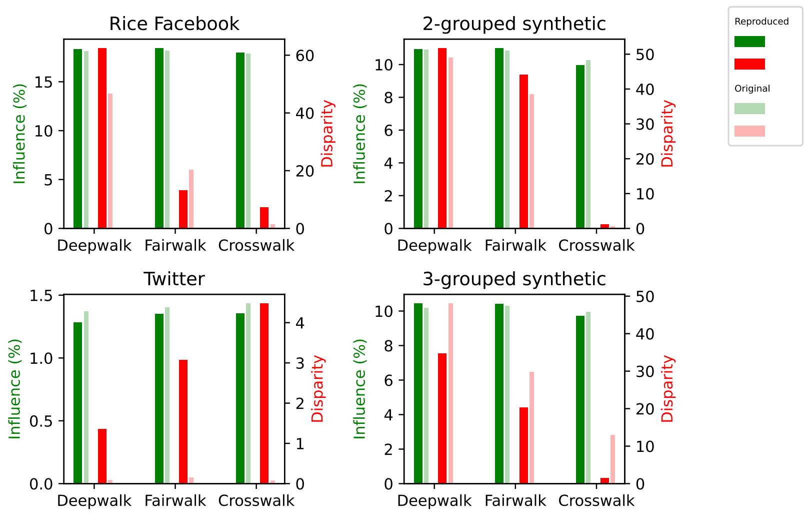


Figure 2. Influence Maximization on datasets used in original paper

Additionally, we apply FairWalk and CrossWalk on Node2Vec, with $p = q = 0.5$, on the Rice Facebook data and observe that CrossWalk outperforms FairWalk again as it experiences a larger decrease in disparity, see results in Figure 4

Result 2 — The results of Khajehnejad et al. on node classification are reproducible. Showed in Figure 4, CrossWalk outperforms FairWalk using Node2Vec as a baseline as was the case in the original paper and confirms statement 2.

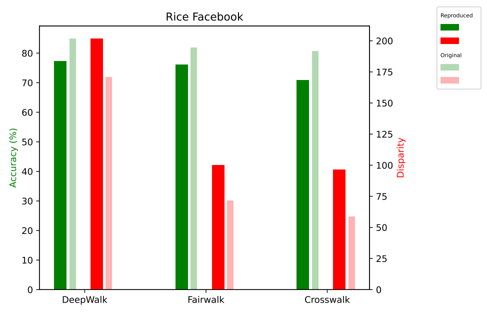


Figure 3. Node classification on the Rice Facebook dataset

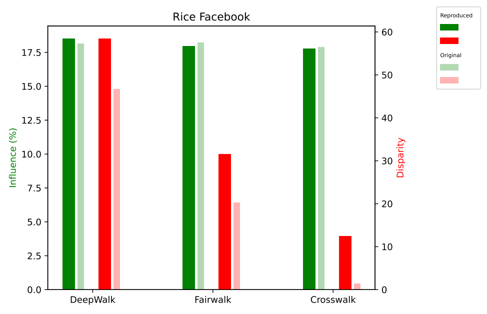


Figure 4. Influence maximization on embeddings obtained by Node2Vec

Result 3 — The performance of embeddings generated by DeepWalk, FairWalk, and CrossWalk is evaluated on link prediction. From Figure 5 can be concluded that we were unable to reproduce the results from Khajehnejad et al. For both datasets, we can't substantiate the claim made by Khajehnejad et al. that CrossWalk outperforms FairWalk as it reduces disparity at the expense of only a slight decrease in accuracy. Hence we are unable to verify statement 3.

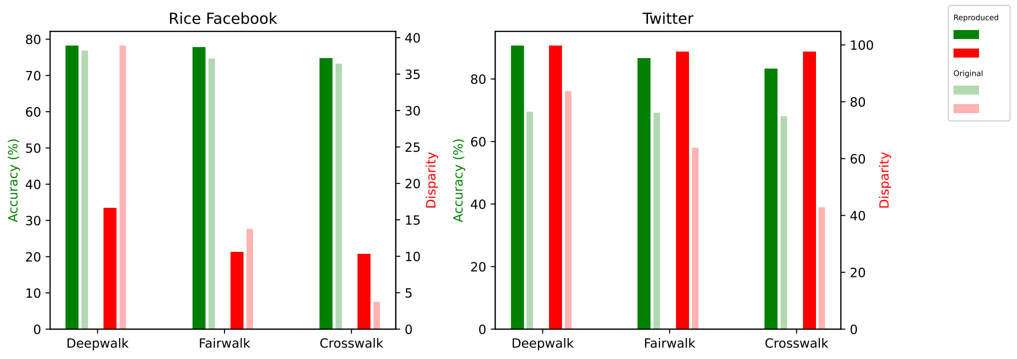
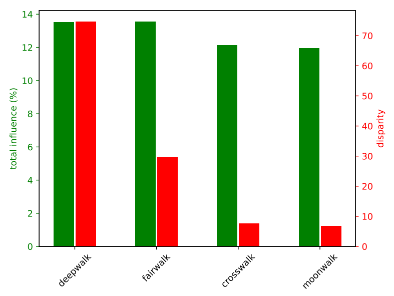


Figure 5. Link prediction on Rice-Facebook and Twitter datasets

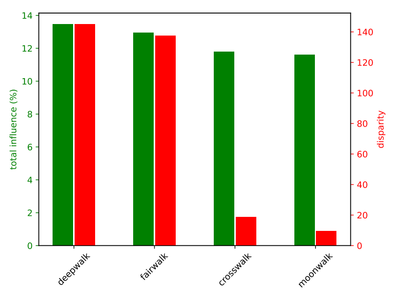
4.2 Results beyond original paper

Additional Result 1 — Figure 6 shows the influence maximization on a 3-grouped, 5-grouped, and 10-grouped synthetic graph. Overall, CrossWalk's superiority is preserved which verifies CrossWalk's ability to generalize as stated in statement 4. The results using a constant group size are shown in sub-figure 6c whereas sub-figure 6d shows results using different group sizes. Notable is that the gap in performance is much higher for FairWalk than it is for CrossWalk. Presumably, this is due to the fact that when group sizes become more equal, a larger fraction of nodes lives on the boundary rather than in the interior of a large group and thus FairWalk is impacted more.

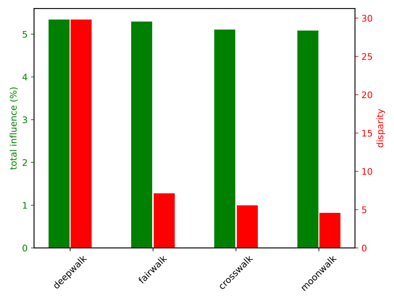
Additional Result 2 — Figure 6 shows the performance of influence maximization on a 3-grouped, 5-grouped, and 10-grouped synthetic graph for the proposed method MoonWalk using p-norm 3. For all graphs, MoonWalk is able to outperform CrossWalk in decreasing disparity while preserving the total influence.



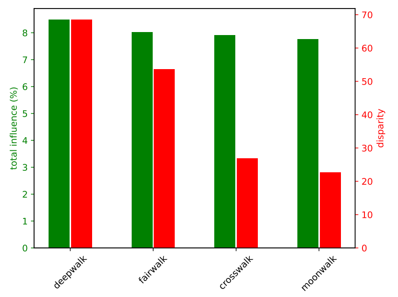
(a) 3-grouped graph with a constant group size of 300, 125, 75



(b) 5-grouped graph with constant group size of 350 (1x),125 (2x) 75 (2x)



(c) 10-grouped graph with a constant group size of 150



(d) 10-grouped graph with group sizes of 400 (1x), 150 (4x) 100 (5x)

Figure 6. Influence maximization on synthetic graphs containing more groups

5 Discussion

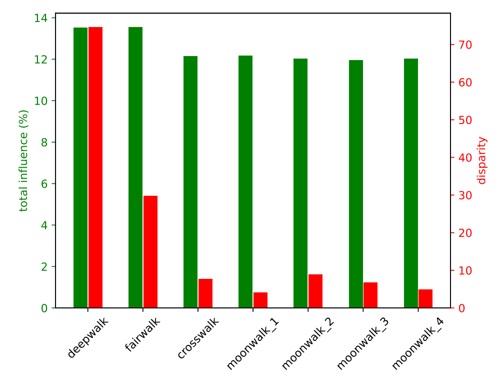
We were pleased to reproduce a significant portion of the original paper. Through a rigorous examination of the results, we managed to formally question the original findings. However, the reliability and credibility of these findings remain uncertain, given the stochastic nature of the random walks and Node2Vec algorithms and the lack of specification of random seeds and hyperparameters by the authors. In our belief, investing time in identifying the appropriate hyperparameters could have facilitated the reproduction of, for instance, the link prediction results. Although the hyperparameters were not specified by the authors, a comprehensive grid search could have enabled their retrieval. Unfortunately, time constraints prevented us from pursuing this aspect in this project, and will be addressed in future work. The same issue was encountered in the search for a suitable Twitter dataset. The Twitter dataset as described in the paper differed significantly from the one found in the provided source code. Given the significant discrepancy in the reproducibility results on the Twitter dataset, it is probable that the difference in this dataset was responsible for the varying results.

Moreover, the implementation of the MoonWalk algorithm produced noteworthy results. As anticipated, MoonWalk demonstrated superior performance when the number of groups increased. Figure 7 in the Appendix also presents intriguing results when different values of the P-norm are considered. Further investigation is necessary to confirm the superiority of the MoonWalk with regard to disparity without sacrificing performance. Overall, theoretical concepts were explained comprehensibly which made understanding easy compared to the technical implementation that was deficiently documented. The authors were unable to respond in time for elaborating on certain implementation details. However, we did receive additional data which was crucial to obtaining certain results.

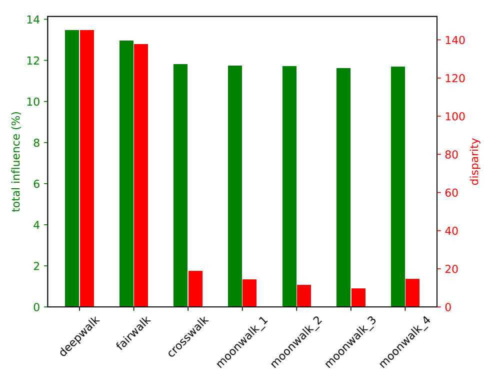
References

1. A. Khajehnejad, M. Khajehnejad, M. Babaei, K. P. Gummadi, A. Weller, and B. Mirzasoleiman. "Crosswalk: Fairness-enhanced node representation learning." In: **Proceedings of the AAAI Conference on Artificial Intelligence**. Vol. 36. 11. 2022, pp. 11963–11970.
2. T. Rahman, B. Surma, M. Backes, and Y. Zhang. "Fairwalk: Towards fair graph embedding." In: (2019).
3. A. Grover and J. Leskovec. **node2vec: Scalable Feature Learning for Networks**. 2016. DOI: 10.48550/ARXIV.1607.00653. URL: https://arxiv.org/abs/1607.00653.
4. T. Mikolov, K. Chen, G. Corrado, and J. Dean. **Efficient Estimation of Word Representations in Vector Space**. 2013. DOI: 10.48550/ARXIV.1301.3781. URL: https://arxiv.org/abs/1301.3781.
5. T. Rahman, B. Surma, M. Backes, and Y. Zhang. "Fairwalk: Towards Fair Graph Embedding." In: **Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19**. International Joint Conferences on Artificial Intelligence Organization, July 2019, pp. 3289–3295. DOI: 10.24963/ijcai.2019/456. URL: https://doi.org/10.24963/ijcai.2019/456.
6. P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo. "The Independent Cascade and Linear Threshold Models." In: Jan. 2015, pp. 35–48.
7. W. Chen, Y. Wang, and S. Yang. "Efficient influence maximization in social networks." In: **Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining**. 2009, pp. 199–208.
8. M. Hardt, E. Price, and N. Srebro. "Equality of Opportunity in Supervised Learning." In: **CoRR** abs/1610.02413 (2016). arXiv:1610.02413. URL: http://arxiv.org/abs/1610.02413.
9. A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel. "You are who you know: inferring user profiles in online social networks." In: **Proceedings of the third ACM international conference on Web search and data mining**. 2010, pp. 251–260.
10. M. Babaei, P. Grabowicz, I. Valera, K. P. Gummadi, and M. Gomez-Rodriguez. "On the Efficiency of the Information Networks in Social Media." In: **Proceedings of the Ninth ACM International Conference on Web Search and Data Mining**. ACM, Feb. 2016. DOI: 10.1145/2835776.2835826.

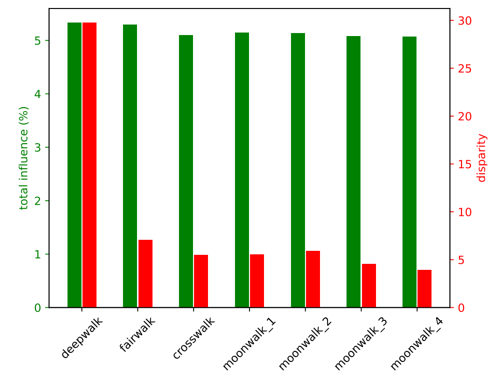
6 Appendix



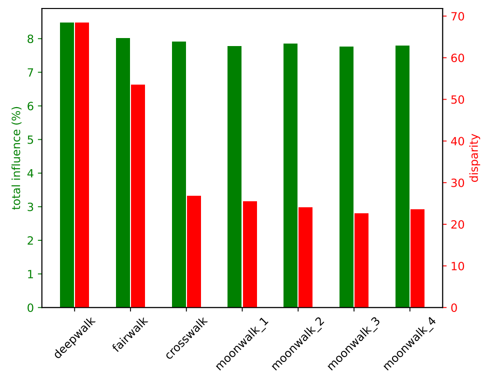
(a) 3-grouped graph with a constant group size of 300, 125, 75



(b) 5-grouped graph with constant group size of 350 (1x),125 (2x) 75 (2x)



(c) 10-grouped graph with a constant group size of 150



(d) 10-grouped graph with group sizes of 400 (1x), 150 (4x) 100 (5x)

Figure 7. Influence maximization on synthetic graphs containing more groups