RESCIENCE C

Replication / ML Reproducibility Challenge 2022

[Re] CrossWalk: Fairness-enhanced Node Representation Learning

Luca Pantea[1, ID] and Andrei Blahovici[1, ID]
[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.8173749

Reproducibility Summary

Scope of Reproducibility – This work aims to reproduce the findings of the paper "Cross-Walk: Fairness-enhanced Node Representation Learning" [1] by investigating the two main claims made by the authors about CrossWalk, which suggest that (i) CrossWalk enhances fairness in three graph algorithms, while only suffering from small decreases in performance, and that (ii) CrossWalk preserves the necessary structural properties of the graph while reducing disparity.

Methodology – The authors made the CrossWalk repository available, which contained most of the datasets used for their experimentation, and the scripts needed to run the experiments. However, the codebase lacked documentation and was missing logic for running all experiments and visualizing the results. We, therefore, re-implement their code from scratch and deploy it as a python package which can be run to obtain all the showcased results.

Results – Our work suggests that the first claim of the paper, which states that Crosswalk minimizes disparity and thus enhances fairness is partially reproducible, and only for the tasks of Node classification and Influence maximization as the parameters specified in the paper do not always yield similar results. Then, the second claim of the paper which states that Crosswalk attains the necessary structural properties of the graph is fully reproducible through our experiments.

What was easy – The original paper contained the necessary information about hyperparameters, which coupled with the publicly available repository made it straightforward to refactor the code and understand the idea of the proposed method.

What was difficult – The difficulty stems from the lack of structure and documentation in the provided code which made the original experiments hard to reproduce. Furthermore, there were missing files in the provided datasets. Also, some experiments were not reproducible at all through the provided code. One more important aspect is that the experiments are CPU intensive which made the reproducibility even harder.

Communication with original authors – Albeit rather late, the authors provided meaningful feedback on our questions about implementation details and initial results.

Copyright © 2023 L. Pantea and A. Blahovici, released under a Creative Commons Attribution 4.0 International license.
Correspondence should be addressed to Luca Pantea (luca.pantea@student.uva.nl)
The authors have declared that no competing interests exist.
Code is available at https://github.com/Dawlau/FACT-AI – DOI 10.5281/zenodo.7932857. – SWH swh:1:dir:6367a5e042f3b14cd6d86349199e08597fc0f3f6.
Open peer review is available at https://openreview.net/forum?id=tpk45Zll8eh.

1 Introduction

The increasing use of machine learning algorithms has led to concerns about their potential to amplify social inequalities and unfairness [2]. To address this, researchers have been working on developing algorithmic tools to detect and mitigate such unfairness [3] [4]. In the reviewed study, the authors propose a new method called CrossWalk, which aims to improve the fairness of various graph algorithms, namely influence maximization, link prediction, and node classification, when applied to node embeddings. CrossWalk is a general method that can be applied to any random walk-based node representation learning algorithm, such as DeepWalk and Node2Vec.

This work aims to address the following goals:

1. **[Reproducibility Study] Reproducing the results from the original paper:** We were able to partially reproduce the claim that Crosswalk enhances fairness, namely for the tasks of node classification and influence maximization. The second claim, which is that the proposed method preserves the higher-order proximity of the graph was successfully reproduced.
2. **[Extended Work] Improvement of the original code:** The original code is not easily runnable, so we had to refactor it and implement our own Bash and Python scripts such that the experiments can be more easily reproduced. Further, we provide a master Python script that allows for the reproducibility of the experiments presented in this report by running only a single terminal command.
3. **[Extended Work] Ablation study:** CrossWalk does not usually yield the expected results out of the box, but by carefully picking the right hyperparameters we managed to make CrossWalk behave as presented in the original paper.
4. **[Extended Work] CrossWalk visualization:** We perform additional visualizations of the edge reweighting procedure and random walk trajectories to investigate the claims proposed by the original authors.
5. **[Proposed Enhancement] Soft Self-Avoiding CrossWalk:** We propose an extension to the algorithm proposed by Khajehnejad et al. [1] which yields better node representations and higher fairness.

2 Scope of reproducibility

This work aims to investigate the reproducibility of the original paper by Khajehnejad et al. [1], which addresses the problem of assessing and mitigating unfairness in graph algorithms by introducing fairness-enhanced node representation learning. Enhancing fairness in machine learning algorithms is receiving growing attention and is becoming an active topic in Ethical Machine Learning [5] [6]. The authors propose a novel edge reweighting method that enhances fairness by biasing random walks initiated in a given group towards visiting nodes on the group boundary and eventually crossing the boundaries between groups. For a detailed overview of the methodology, datasets, and metrics used by the authors, we invite the reader to consult Section 3.

The **main claims** proposed in the original paper can be summarised as follows:

1. **Fairness-enhanced node representational learning**. The authors claim that by applying CrossWalk to learn node representations through any stochastic traversal algorithm (such as DeepWalk [7] or Node2Vec [8]), disparity values significantly decrease for Node Classification, Influence Maximization, and Link Prediction, while only suffering from small decreases in accuracy.

2. **Preserving higher-order proximity of the graph**. The claim is that CrossWalk is able to preserve the necessary structural properties of the graph while bringing peripheral nodes towards neighboring nodes from other groups in the embedding space.

In addition to reproducing the results showcased in the original paper, we perform further experimentation in order to validate the robustness of the algorithm and to test the claimed effectiveness of the methodology in producing representations with higher fairness and smaller discrepancy between different groups.

3 Methodology

The original CrossWalk implementation is publicly available in their GitHub repository. However, not all experiments are perfectly reproducible based on the author's code, therefore we had to make minor adjustments to how we use CrossWalk for our research. Furthermore, we build upon the work of Khajehnejad et al. [1] by reorganizing the code to make our experiments more reproducible and by providing supplementary updates to CrossWalk.

3.1 Model description

CrossWalk is a re-weighting method that is used to bias random walk-based algorithms towards visiting multiple groups, which in turn enhances fairness. This is mainly done by re-weighting the probabilities associated with the graph edges as follows:

1. CrossWalk increases the weights of edges **near the group boundaries**.
2. CrossWalk increases the weights of edges **that connect nodes from different groups**.

The mathematical formulas that describe the re-weightings from points 1 and 2 are the following:

$$w'_{vu} = \begin{cases} w_{vu}(1-\alpha) \times \frac{m(u)^p}{\sum\limits_{z \in N_v} w_{vz} m(z)^p} & \text{if } u \in N(v), l_v = l_u \\ w_{vu}\alpha \times \frac{m(u)^p}{|R_v| \times \sum\limits_{z \in N_v^c} w_{vz} m(z)^p} & \text{if } u \in N(v), l_v \neq l_u = c. \end{cases} \tag{1}$$

where w_{vu} is the initial weight of the edge (u, v), α and p are hyperparameters, R_v the set of groups in the neighbourhood of v, l_x the group that node x is part of, and $m(z)$ is the proximity of a node as described in the original paper. We have also elaborated in Appendix A the algorithms that are relevant towards fully understanding the ideas presented in this report.

3.2 Datasets

We follow the outline of the original paper and work towards reproducing the experiments supporting each of the two claims. We present the relevant datasets along with further information related to them, including the underlying task using the dataset, sample size and distribution, and a description in Table 1.

Dataset	Num. samples	Num. groups	Samples/Group	Description
Rice-Facebook	441	2	344/97	Social relations
Twitter	3560	3	2598/782/180	Political learning
2-Grouped Synthetic	500	2	350/150	Synthetic network
3-Grouped Synthetic	500	3	300/125/75	Synthetic network

Table 1. Summary of the datasets used in our experimentation. For each of the tasks, the following train/test ratios are used: **Node Classification**: 0.5 (Rice), **Link Prediction**: 0.1 (Rice Twitter), **Influence Maximization**: n/a (Rice 2,3-Grouped Synthetic, Twitter).

3.3 Hyperparameters

To reproduce the results of the paper, we identify and primarily use the same hyperparameters as in the original paper. However, we set out to further investigate the robustness of the originally proposed approach by performing ablation studies for the parameters listed in Table 2 for each task.

Task	α	p
Node Classification	[0.1, 0.3, **0.5**, 0.7, 0.9]	[**1**, 2, 4, 6, 8]
Link Prediction	[0.1, 0.3, **0.5**, 0.7, 0.9]	[1, **2**, 4, 6, 8]
Influence Maximization	[0.1, 0.3, **0.5**, **0.7**, 0.9]	[1, **2**, **4**, 6, 8]

Table 2. Summary of the hyperparameter values used in our experimentation. The parameter values in bold font represent the default values within the original experiments.

3.4 Experimental setup and code

This work presents a restructured version of the CrossWalk repository, a collection of Python-based code for performing various graph-based tasks. Our main contribution is the development of shell files that can be used to reproduce all experiments on a given device, with parameters specified by the user. The most commonly used metrics for evaluation are accuracy as a percentage and disparity, which is used to measure the fairness between different groups. The disparity is calculated as the variance of the performance of a model (Q) between different groups (C).

$$disparity(A) = Var(\{Q_i\} : i \in [C]) \tag{2}$$

The code is provided in the GitHub repository FACT-AI. We acknowledge that not anybody might have access to the required computational resources to generate the necessary embeddings, and thus we provide them in the HuggingFace repository fact-ai.

3.5 Computational requirements

We perform all experiments locally, using an AMD Ryzen 7 5800H CPU, with 16 threads and an Apple M1 Max chip with 10 CPU cores. For training the adversarial autoencoder, an Nvidia GeForce RTX 3080 GPU was used. The total computational cost for running all experiments comes at a total of roughly 160 CPU hours and 10 GPU hours.

4 Results

The results reproduced from the original paper show that CrossWalk does not usually yield the expected results out of the box, but rather it is highly sensitive to the choice of the hyperparameters α and p. In Appendix D, we provide an extensive ablation study of the hyperparameters and explain what impact they have on the performance of CrossWalk. While this is true for the tasks of node classification and influence maximization, for link prediction CrossWalk seems to be showing worse results than Fairwalk and vanilla DeepWalk using the provided source code. One important mention is that all of the results of the experiments involving the 3 graph algorithms presented are obtained by averaging the performance across 5 distinct runs.

4.1 Results reproducing original paper

Claim 1: Fairness-enhanced node representational learning — *Partially correct*
Table 3 displays an overview of all tasks, data sets, and their reproducibility.

Task	Dataset	Reproducible?
Visualization	Rice-Facebook	✓
IM	Rice-Facebook	✓
IM	2-grouped synthetic	✓
IM + Node2Vec	Rice-Facebook	✗
IM	3-grouped synthetic	✓
IM	Twitter	✓
LP	Rice-Facebook	✗
LP	Twitter	✗
NC	Rice-Facebook	✓

Table 3. Overview of the performed experimentation. ✓- that similar values were obtained, while ✗- the values obtained did not match the paper.

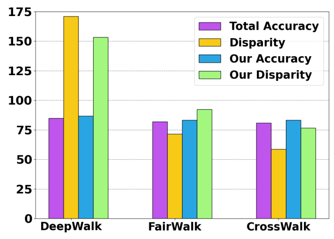
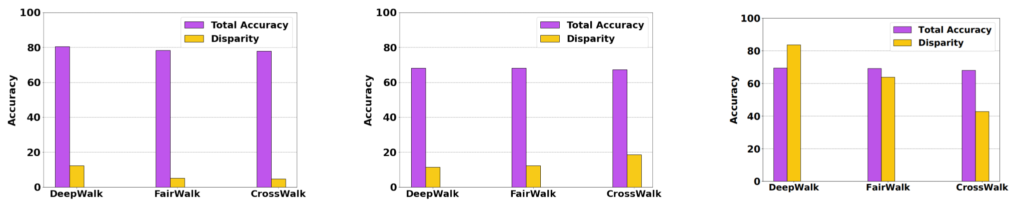


Figure 1. Node classification accuracy and disparity on the Rice-Facebook Dataset ($\alpha = 0.5, p = 1$)(Sub-figure a), comparing our results against the CrossWalk's authors.

As we already mentioned in the introduction of the Results section, the results for **link prediction** were not reproducible using the original source code, and this is shown in Figure 2.



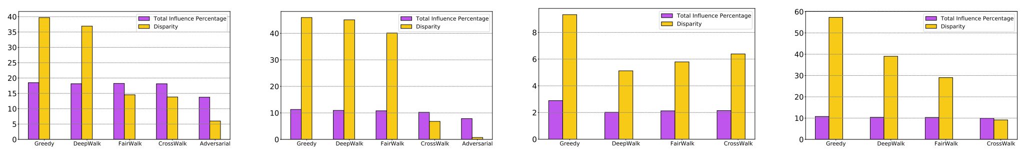
(a) Our results - Rice-Facebook ($\alpha = 0.5, p = 2$)

(b) Our results - Twitter ($\alpha = 0.5, p = 2$)

(c) Khajehnejad et al.'s [1] results - Twitter ($\alpha = 0.5, p = 2$)

Figure 2. Link prediction accuracy and disparity on the Rice-Facebook Dataset (Sub-figure a) and Twitter Dataset (Sub-figure b) for DeepWalk, Fairwalk, and CrossWalk, and the CrossWalk's authors results in Sub-figure c) for comparison.

Figure 3 shows the attempt to reproduce the original results of influence maximization experiments. The methods used were a Greedy algorithm baseline, Deepwalk with node re-weighting, Fairwalk, and Crosswalk. Crosswalk was expected to show the lowest disparity, but this was not the case, and it was shown in the Appendix D that choosing the right hyperparameters is crucial. Overall, the results were reproducible, but hyperparameter tuning was necessary. Unfortunately, no code was provided for the Node2Vec experiments, so they could not be reproduced.

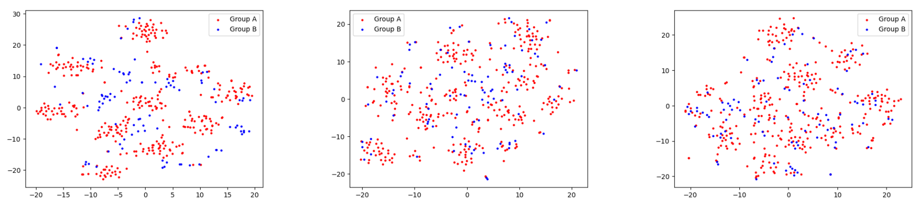


Rice-Facebook $\alpha = 0.5,\ p = 4$ | 2-grouped synthetic $\alpha = 0.7,\ p = 4$ | Twitter $\alpha = 0.5,\ p = 2$ | 3-grouped synthetic $\alpha = 0.7,\ p = 4$

Figure 3. Influence maximization disparity and total influence percentage on the Rice-Facebook Dataset (Sub-figure 1), 2 and 3-grouped synthetic datasets (Sub-figure 2 and 4) and the Twitter Dataset (Sub-figure 3) for the Greedy algorithm, DeepWalk, Fairwalk, CrossWalk and an Adversarial autoencoder [9].

Moreover, we were successful in reproducing the node classification experiment on Rice-Facebook using the provided hyperparameters α and p. As we can see from Figure 1, the results are similar to the original paper, only with some really small differences in the actual disparity values.

Claim 2: Preserving high-order proximity of the graph — *Correct*
Our work shows that CrossWalk is able to preserve the structural properties of the graph while bringing nodes from different groups closer together in the embedding space. Figures 4(a-c) demonstrate this, as peripheral nodes from two groups are closer together after applying edge re-weighting.



(a) Rice-Facebook Dataset DeepWalk
(b) Rice-Facebook Dataset CrossWalk ($\alpha = 0.5, p = 2$)
(c) Rice-Facebook Dataset CrossWalk ($\alpha = 0.5, p = 4$)

Figure 4. Distribution of the embedded nodes from the two groups generated by DeepWalk - Subfigure a), and with edge re-weighting b) and c) .

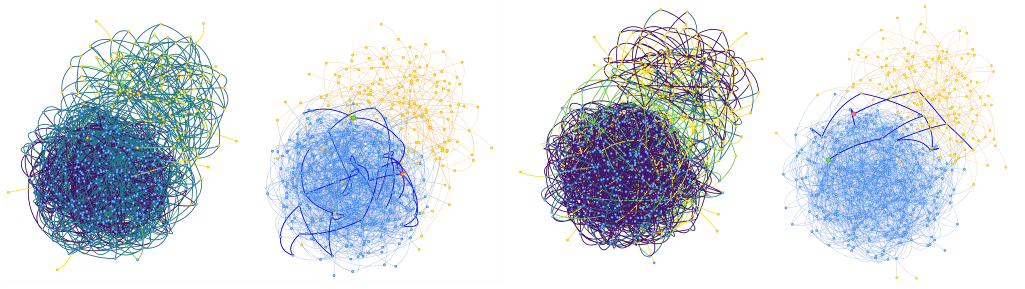
4.2 Results beyond original paper

Ablation study on the CrossWalk hyperparameters — We have conducted an ablation study to see how different combinations of the hyperparameters α and p (Table 3.3) affect the results in the influence maximization and node classification tasks. This study has been elaborated more in Appendix D and it was pivotal towards reproducing the paper as it allowed us to see that choosing the right pair of hyperparameters allows Crosswalk to outperform all of the other random walk-based algorithms.

Explainability analysis of CrossWalk — The experiments outlined above (Figure 4) replicating the results of the original paper offer a certain degree of evidence for the second claim, primarily by visual comparison of the DeepWalk projection of the node embeddings both using the weighted (via CrossWalk) and unweighted graphs. However, these qualitative experiments do not offer a clear overview of the modified edge distribution, the behaviour of biased random walks, and the correlation to a decrease in accuracy to support the second claim. We, therefore, carry out a more in-depth analysis by **combining** both **quantitative** and **qualitative analysis** of CrossWalk, as depicted in Figure 5, and in Table 4, which will serve as the motive behind our proposed extension.

Motivation for extension of CrossWalk: CrossWalk's re-weighting mechanism achieves the desired behaviour of increasing the edge weights near and on the group boundaries. However, this leads to **undesired random walk trajectories**, where the random walks initiative by the algorithm appear to travel back and forth between the edges located at group peripheries. This is an important limitation to take into consideration, as it **limits the ability to capture structural information and retain higher-order proximity** between nodes. From our experimentation, we observe that throughout training, CrossWalk visits about **30% fewer distinct nodes** than DeepWalk. Our stated considerations about the limitations of CrossWalk motivate the need for an extension of the original algorithm, which stimulates the random walk to avoid ineffective path trajectories.

Soft Self-Avoiding CrossWalk — Upon further examining CrossWalk's random walk trajectories (Figure 5), we encountered an issue not mentioned in the original paper. During



(a) Graph edge weights (Left) DeepWalk random walk (Right)

(b) Re-weighted graph edge weights (Left) CrossWalk random walk (Right)

Figure 5. Explainability analysis of CrossWalk: Sub-figures a) and b) present the differences between the two methods, DeepWalk and CrossWalk, respectively on the 2-grouped Synthetic dataset. Node color determines the group. Lighter edge colors represent higher weights, while darker colors indicate smaller weights. Nodes colored in green and red indicate the source and destination of a given path, respectively.

	Synthetic 2		Synthetic 3		Rice-Facebook		
Methods	*avg. cross.*	*% vis.*	*avg. cross.*	*% vis.*	*avg. cross.*	*% vis.*	*Acc**
DeepWalk	1.6	80.23%	1.69	74.12%	6.66	92.58%	86.66%
CrossWalk	11.9	64.7%	11.89	57.42%	18.86	78.74%	83.32%
SSA CrossWalk	**9.15**	**85.68%**	**8.6**	**81.22%**	**18.59**	**91.58%**	**84.96%**

Table 4. Quantitative comparison of the discussed node representation methodologies on three datasets. The average number of crossings (*avg. cross.*) indicates the mean number of times the random walk has crossed a group boundary throughout training. The visited percentage (*% vis.*) represents the fraction of distinct nodes visited in a path throughout training. Finally, the *node classification* accuracy is reported for each method for the Rice-Facebook dataset. Our proposed method is highlighted with a heavier (bolded) font.

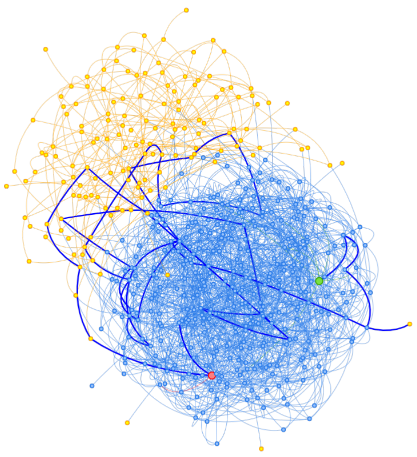
the edge re-weighting process, the edges bordering or crossing group boundaries would attain significantly higher values in comparison to other edges, thus biasing the random walk to recurrently traverse a limited number of edges. As a result, the **structural properties of the graph are not fully utilized**, leading to **decreased representation quality**. This can be observed both in Figure 5 and in Table 4, where the number of distinct visited nodes is significantly smaller than DeepWalk.

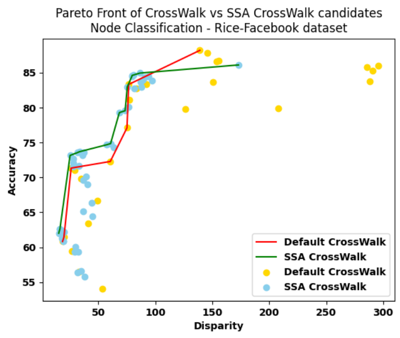
Therefore, we have proposed a solution based on Self-Avoiding Walks [10] (Background in Appendix B) which steers the random walks towards avoiding previously visited edges by using a discounting function, parameterized by γ, as presented in Equation 3.

$$P(u_i = u | u_{i-1} = v) = \begin{cases} \pi_{uv} \cdot \gamma^{c(u,v)} & \text{if } (u,v) \in E \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Where $c(u,\ v)$ stores the number of times the walk has traversed the edge between node u and v. We provide an overview of the space and time complexity in Appendix C. As $\gamma \in [0,\ 1]$, if $\gamma \to 1$, the behaviour is similar to CrossWalk, while $\gamma \to 0$ stimulates the random walk to avoid already visited edges. We perform initial trials with larger values of γ, yet observe a meaningful effect only in the range $[0.2,\ 0.4]$. We carry out experiments with the entire hyperparameter space of CrossWalk (Section 3.3), with added $\gamma \in [0.2, 0.3, 0.35, 0.4]$, showcase the results in Figures 6 and 7, and provide performance metrics in Table 4.

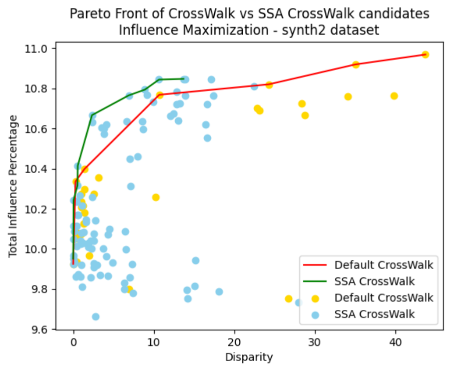
The results indicate that the proposed extension **consistently outperforms** the original implementation of CrossWalk on the two reproducible tasks - Link Prediction and Node Classification. We validate the effectiveness of our extension via Figures 6b and 7, where we demonstrate that the Soft Self-Avoiding CrossWalk **not only achieves better accuracy and influence percentage** but **additionally minimizes the disparity** (thus maximizing fairness) for each task.


(a) SSA CrossWalk random walk

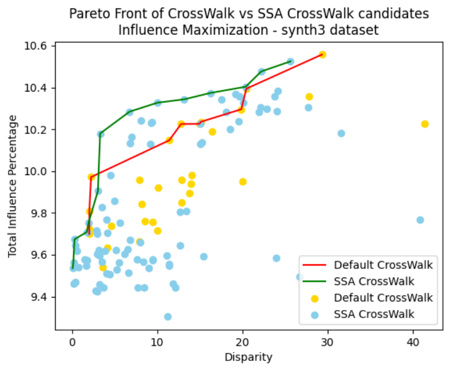


(b) Pareto optimal solutions for SSA CrossWalk vs Default CrossWalk for Node Classification.

Figure 6. **Soft Self-Avoiding CrossWalk**: A visualization of the Soft Self-Avoiding random walk is presented in Sub-figure a), and the individual Node classification results for all combinations of hyperparameters are plotted in Sub-figure b). The Pareto front is highlighted for each method (The utopian objective is Top-Left).



(a) Pareto optimal solutions for SSA CrossWalk vs Default CrossWalk on the 2-grouped synthetic dataset



(b) Pareto optimal solutions for SSA CrossWalk vs Default CrossWalk on the 2-grouped synthetic dataset

Figure 7. **Soft Self-Avoiding CrossWalk**: Influence maximization - comparison analysis of the proposed method for all hyperparameter configurations on the 2-grouped and 3-grouped synthetic datasets (Utopian objective is Top-Left).

5 Discussion

In this work, several experiments were conducted in order to attempt to reproduce the findings of CrossWalk [1]. Most of the experiments in the original CrossWalk paper were **feasible to reproduce**. The claims were found to be **largely valid**, however, **with a few caveats**.

The first claim states that **CrossWalk results in a lower disparity** for all three graph algorithms investigated. Our findings show that CrossWalk was **successful in reducing disparity** in two of the three graph algorithms investigated, namely **influence maximization and node classification**.

The second claim, which states that CrossWalk preserves the higher-order proximity of the graph, was found to be supported. We expanded our research methodology to include an explainability analysis to further demonstrate the validity of the claim. The results of the explainability analysis showed an **undesired behavioural property of CrossWalk**, where the random walk goes back and forth between edges on the group peripheries, causing the algorithm to **visit on average 30% fewer nodes**. This finding prompted the creation of the **SSA CrossWalk**, a method incorporating a discount function, that aims to stimulate the trajectories of the walks to visit more distinct edges. The results obtained show that SSA CrossWalk increases accuracy and decreases disparity independent of the hyperparameter configurations used for the original algorithm, as shown in Figure 7.

Limitations: Our contributions suffer from three main drawbacks. First, due to time constraints, we were unable to resolve the outstanding issues regarding link predictions, yet we were able to obtain results that follow the same trends for the other tasks. Second, the re-implementation of the original base significantly decelerated the reproduction process. And finally, the proposed approach is independent of the underlying graph characteristics, which encourages future work in the direction of estimating γ via a non-linear function approximation (i.e. neural network-based approaches).

5.1 What was easy

The original paper contained the necessary information about the edge re-weighting mechanism and the values of the hyperparameters, which coupled with the publicly available repository, made it straightforward to refactor the code and understand the idea of the proposed method.

5.2 What was difficult

Despite having access to the original code for CrossWalk, the process of reproducing the results took significantly more time than initially expected due to a lack of comments, poor code structure, and no documentation. Furthermore, obtaining similar results for the reproducible experiments proved to be not trivial due to the time constraints.

5.3 Communication with original authors

The authors cleared up discrepancies in our results and provided additional details through email correspondence. More specifically, we were able to reproduce the results for Node Classification with an additional file provided by the authors, which was initially missing from their repository.

References

1. A. Khajehnejad, M. Khajehnejad, M. Babaei, K. P. Gummadi, A. Weller, and B. Mirzasoleiman. **CrossWalk: Fairness-enhanced Node Representation Learning**. 2021. DOI: 10.48550/ARXIV.2105.02725. URL: https://arxiv.org/abs/2105.02725.
2. D. Pessach and E. Shmueli. "A Review on Fairness in Machine Learning." In: **ACM Comput. Surv.** 55.3 (Feb. 2022). DOI: 10.1145/3494672. URL: https://doi.org/10.1145/3494672.
3. N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. "A Survey on Bias and Fairness in Machine Learning." In: **CoRR** abs/1908.09635 (2019). arXiv:1908.09635. URL: http://arxiv.org/abs/1908.09635.
4. H. Suresh and J. V. Guttag. "A Framework for Understanding Unintended Consequences of Machine Learning." In: **CoRR** abs/1901.10002 (2019). arXiv:1901.10002. URL: http://arxiv.org/abs/1901.10002.
5. M. Sicart, I. Shklovski, and M. Jones. "Can Machine Learning be Moral?" In: **CoRR** abs/2201.06921 (2022). arXiv:2201.06921. URL: https://arxiv.org/abs/2201.06921.
6. I. Y. Chen, E. Pierson, S. Rose, S. Joshi, K. Ferryman, and M. Ghassemi. "Ethical Machine Learning in Health Care." In: **CoRR** abs/2009.10576 (2020). arXiv:2009.10576. URL: https://arxiv.org/abs/2009.10576.
7. B. Perozzi, R. Al-Rfou, and S. Skiena. "DeepWalk: Online Learning of Social Representations." In: **CoRR** abs/1403.6652 (2014). arXiv:1403.6652. URL: http://arxiv.org/abs/1403.6652.
8. A. Grover and J. Leskovec. "node2vec: Scalable Feature Learning for Networks." In: **CoRR** abs/1607.00653 (2016). arXiv:1607.00653. URL: http://arxiv.org/abs/1607.00653.
9. M. Khajehnejad, A. A. Rezaei, M. Babaei, J. Hoffmann, M. Jalili, and A. Weller. "Adversarial Graph Embeddings for Fair Influence Maximization over Social Networks." In: **CoRR** abs/2005.04074 (2020). arXiv:2005.04074. URL: https://arxiv.org/abs/2005.04074.
10. R. Bauerschmidt, H. Duminil-Copin, J. Goodman, and G. Slade. **Lectures on Self-Avoiding Walks**. 2012. DOI: 10.48550/ARXIV.1206.2092. URL: https://arxiv.org/abs/1206.2092.
11. T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient Estimation of Word Representations in Vector Space." In: **Proceedings of Workshop at ICLR** 2013 (Jan. 2013).
12. 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.
13. N. Madras and G. Slade. "The self-avoiding walk." In: 1991.
14. B. Nienhuis. "Exact Critical Point and Critical Exponents of $\mathrm{O}(n)$ Models in Two Dimensions." In: **Phys. Rev. Lett.** 49 (15 Oct. 1982), pp. 1062–1065. DOI: 10.1103/PhysRevLett.49.1062. URL: https://link.aps.org/doi/10.1103/PhysRevLett.49.1062.
15. E. J. J. van Rensburg. "Statistical mechanics of directed models of polymers in the square lattice." In: **Journal of Physics A: Mathematical and General** 36.15 (Apr. 2003), R11. DOI: 10.1088/0305-4470/36/15/201. URL: https://dx.doi.org/10.1088/0305-4470/36/15/201.

A Relevant methods used in the paper

DeepWalk: DeepWalk [7] learns latent representations of nodes in a network G by iteratively initiating truncated random walks starting from randomly sampled positions and updating the node representations by optimizing the Skip-gram likelihood objective as outlined by Mikolov et al. [11] using a hierarchical soft-max.

Fairwalk: Introduced by Rahman et al. [12], Fairwalk is a modified random walk algorithm extended from Node2Vec [8], which creates a more *diverse network neighborhood representation* by optimizing for statistical parity and equality of representation at both user and network level.

Adversarial AutoEncoder: Another proposed method examined by the authors is the Adversarial AutoEncoder [9] to generate node representations instead of random walk-based methods. We make use of the paper's publicly available implementation and make modifications to its implementation to accommodate our input structure.

Greedy baseline: This is a greedy method used for selecting the most influential nodes in a graph based on heuristics, and it does not use node embeddings.

B Background Self-Avoiding Walks

Self-avoiding walks (SAWs) are paths within an d-dimensional integer lattice $\mathcal{Z}^d$, without self-intersections [13]. An n-step self-avoiding walk is defined to be the ordered set of $n+1$ nodes in the $\mathcal{Z}^d$ manifold, in which each consecutive edge does not appear more than once. In the context of the random walk algorithms, self-avoiding walks are useful since they model random movements that do not allow for self-intersection, which can better capture the locality and structural characteristics of a given graph [14]. This property is important for applications where the random walk must avoid certain regions or paths. In such cases, SAWs can be used as a way of generating a random path that satisfies these constraints. For example, in computer simulations of physical and chemical systems, SAWs can be used to model the random movement of a polymer chain or a molecule, avoiding any self-intersection that would represent a physically unrealistic behaviour [15].

C Complexity analysis of Soft Self-Avoiding CrossWalk

Our proposed addition to the CrossWalk algorithm stimulates the random walk towards avoiding recurrently traversing a limited number of edges, by using a discounting function $\gamma^{c(u,v)}$. Our implementation uses a HashMap initialized per random walk for storing the counts of the previously visited edges, which has a worst-case **time and space complexity** of $\mathcal{O}(K)$, where K represents the preset random walk length. Thus, our proposed method scales with K, which **supports scalability**, as $K << |E|$, where E is the set of graph edges for a given graph.

D Ablation Study

The Crosswalk algorithm is highly sensitive to the choice of the hyperparameters α and p as we have already shown in the Results section. Considering this, we decided to do an ablation study on these hyperparameters to see if we are able to make Crosswalk achieve a lower disparity for all tasks. In the figures below, we show the results for each pair of hyperparameters, where the values for α and p respectively are presented in Table 2. As you can see, choosing the right pair of hyperparameters is crucial, as for all the tasks, there exists a pair that yields in the lowest disparity compared to the other methods.

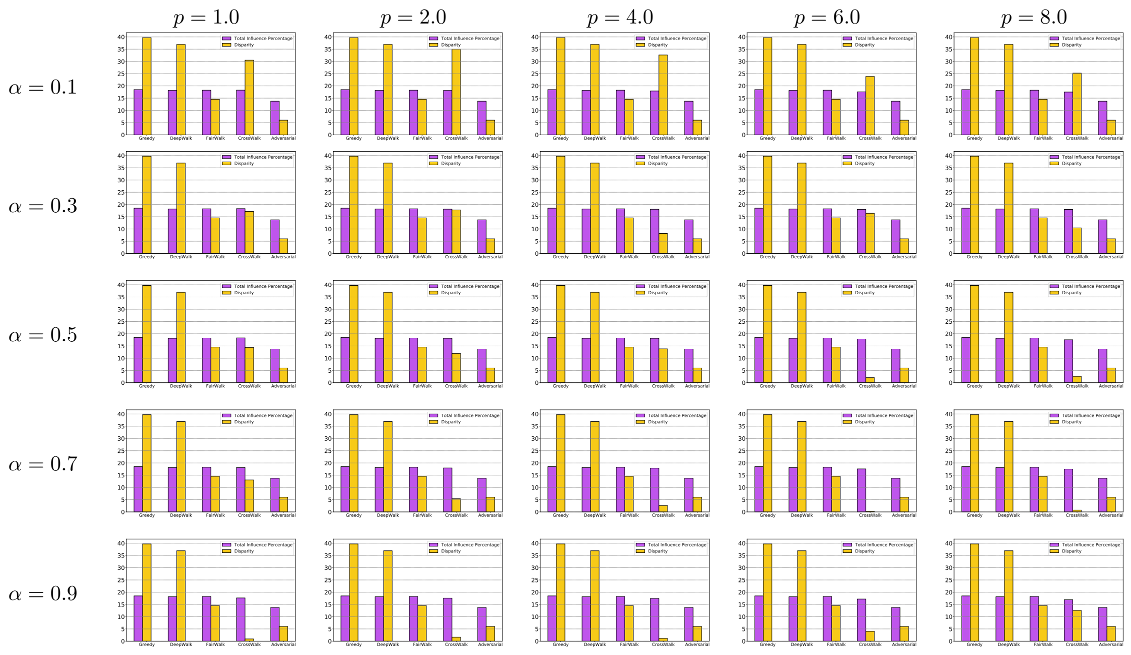


Figure 8. Influence maximization ablation on Rice-Facebook dataset, using the hyperparameters mentioned under 3.3. The yellow bar on the plot represents the disparity metric, while the purple bar indicates the total influence percentage.

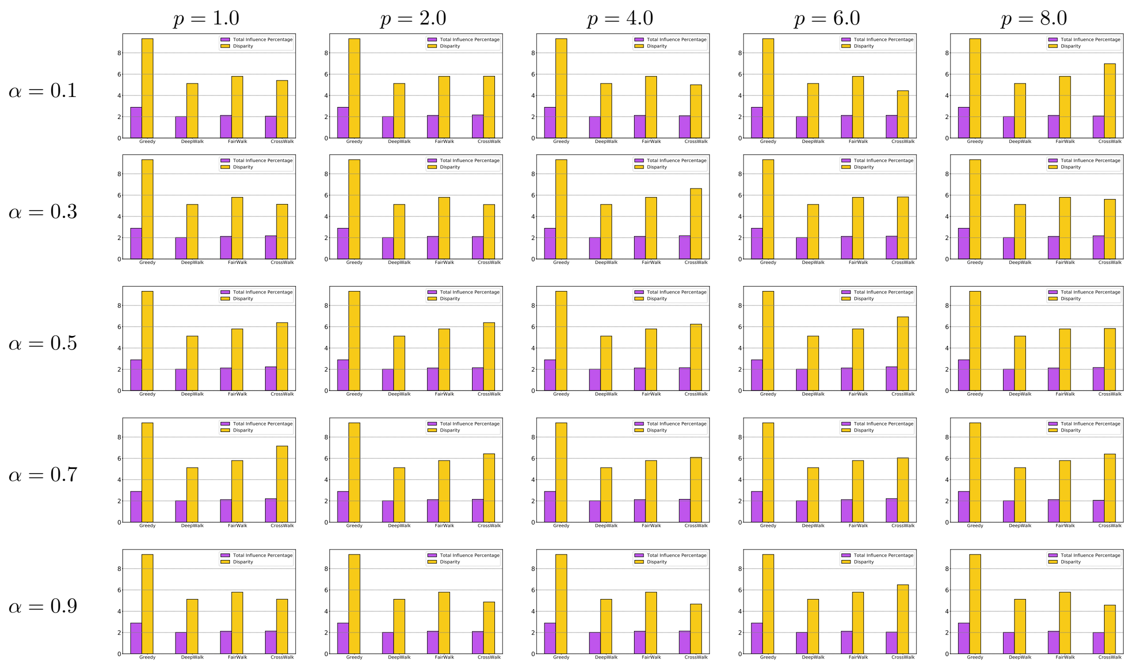


Figure 9. Influence maximization ablation study on the Twitter dataset, using the hyperparameters mentioned under 3.3. The yellow bar on the plot represents the disparity metric, while the purple bar indicates the total influence percentage.

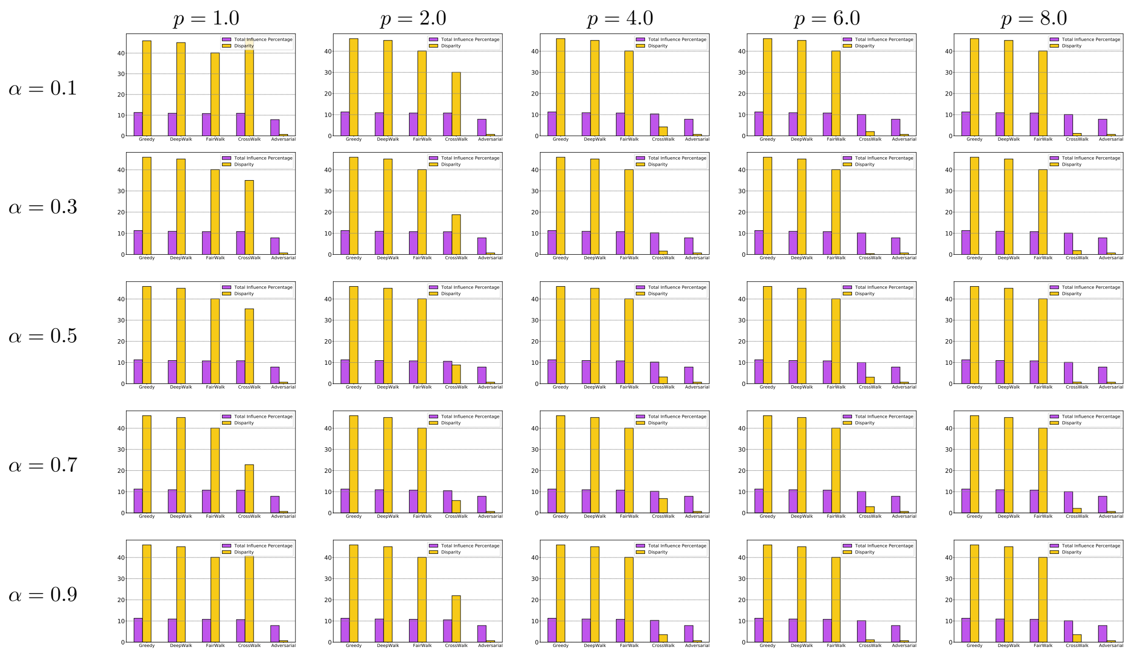


Figure 10. Influence maximization ablation study on the 2-grouped dataset, using the hyperparameters mentioned under 3.3. The yellow bar on the plot represents the disparity metric, while the purple bar indicates the total influence percentage.

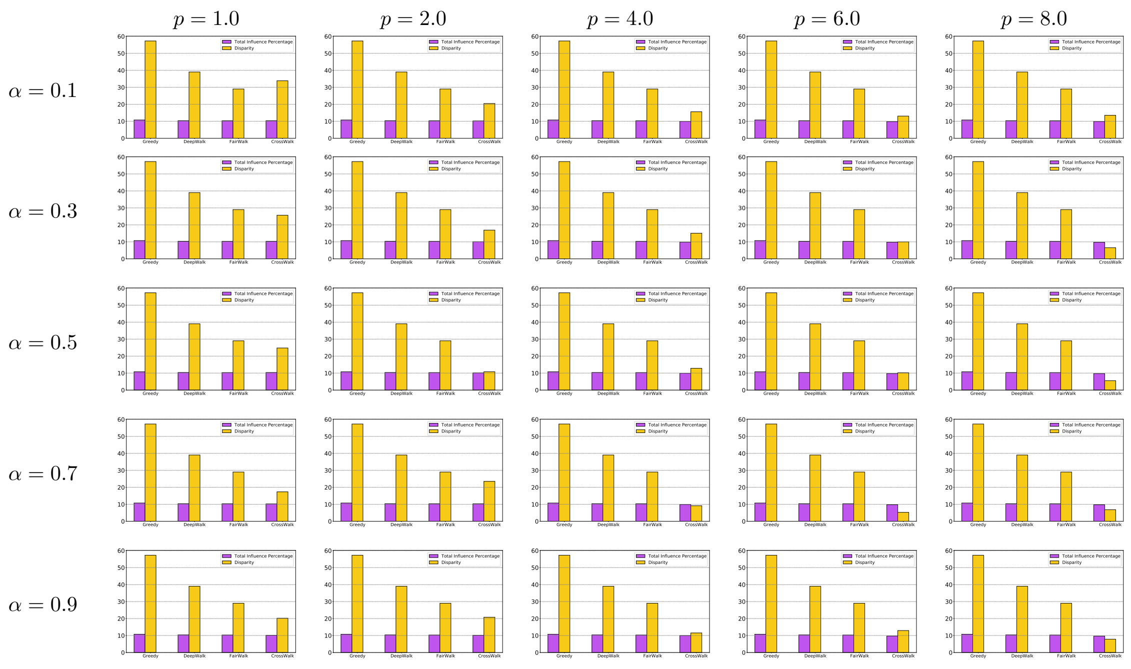


Figure 11. Influence maximization ablation study on the 3-grouped dataset, using the hyperparameters mentioned under 3.3. The yellow bar on the plot represents the disparity metric, while the purple bar indicates the total influence percentage.

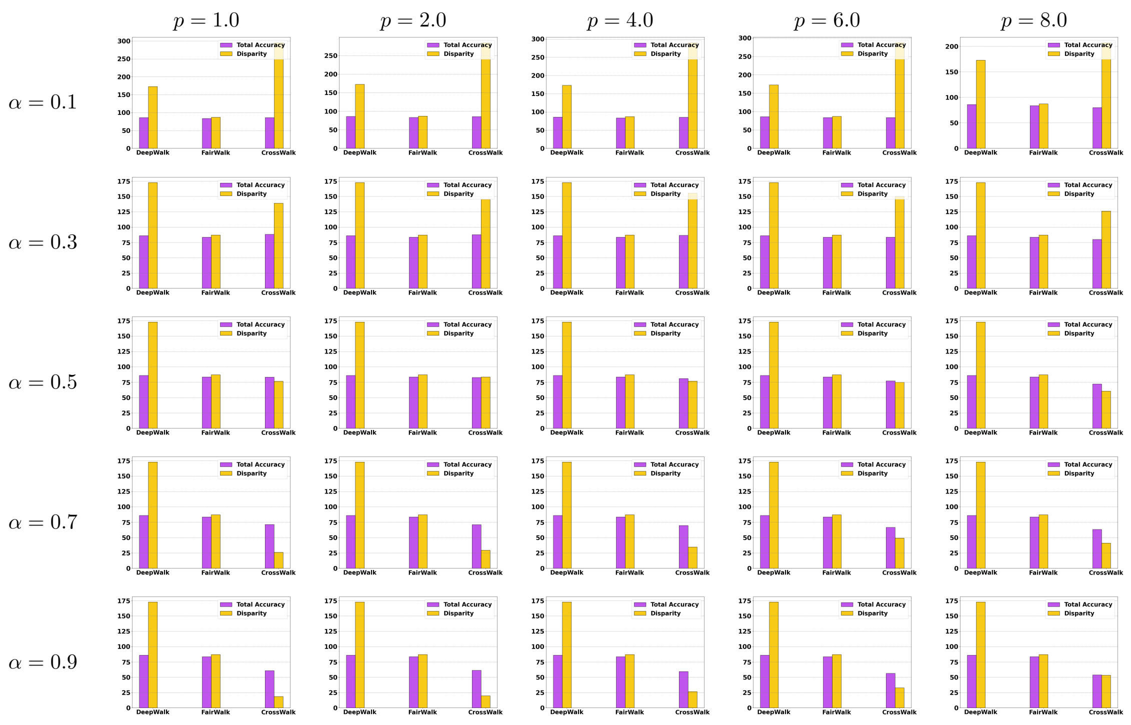


Figure 12. Node classification ablation study on Rice-Facebook dataset, using the hyperparameters mentioned under 3.3. The yellow bar on the plot represents the disparity metric, while the purple bar indicates the total obtained accuracy.