



**Polarisation and Influence in Online Social Networks  
Diversity-Aware Reranking of Node2vec-based Recommendations in Social  
Networks**

**Teodora Mihaila**

**Supervisor(s): Anna Latour**

**<sup>1</sup>EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,  
In Partial Fulfilment of the Requirements  
For the Bachelor of Computer Science and Engineering  
June 21, 2026

Name of the student: Teodora Mihaila  
Final project course: CSE3000 Research Project  
Thesis committee: Anna Latour, Megha Khosla

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

## Abstract

Recent elections have demonstrated how social media contributes to political polarization, leading to real world consequences. Conventional "People you may know" algorithms used for new connection recommendations rely on structural similarity. These algorithms can recommend connections between people with similar opinions. In effect, this may expose users to opinions that reinforce their initial beliefs. Several methods have been introduced to reduce polarization based on interventions in the recommendation algorithm. However, none of the methods relies on post-processing the link scores. We aim to fill this gap by studying whether a diversity-aware reranking of a node2vec-based link prediction can decrease polarisation in synthetic graphs. We compare two reranking methods based on opinion and community diversity against the baseline. We look at prediction quality and polarisation change under two different opinion dynamic models: DeGroot and BCM. Our results indicate a negligible effect on polarization. However, the form of reranking has different effects based on the opinion dynamics model used.

## 1 Introduction

Recent elections have shown how quickly online social networks can contribute to political polarization. Social media platforms allow opinions, narratives, and influence to spread through large groups of users. When users are mainly exposed to content and people who reinforce their existing views, *echo chambers* can form [1].

Online polarization can also have real-world consequences, including protests, hostility, and disruptions of democratic institutions and processes. Examples include the 2018 Brazilian election, during which encrypted messaging platforms were used to spread targeted political misinformation [2], the widespread public dispute after fraud conspiracies flooded social media after the 2020 US presidential election [3], and the annulment of the first round of the 2024 Romanian presidential election following concerns about foreign interference and social media influence [4]. These examples motivate the need to better understand how online networks can amplify or reduce political polarization.

One important part of this problem is the structure of the social graph. People recommendation systems, which suggest potentially relevant social connections to users, can influence the creation of new friendships between users. This can change who users are exposed to and how opinions spread through the network. Previous work has shown that people recommenders can affect echo chambers and polarization, especially when users are already friends with people having similar opinions [5].

Many "People you may know" algorithms are based on link prediction, which aims to infer which new friendships are likely to appear in the future. They often recommend connections between users who are close in the graph, such as users

with shared neighbors [6]. In polarized networks, structural similarity can overlap with opinion similarity. As a result, link prediction methods may mostly recommend links inside the same opinion group.

*Node2vec* is a common graph embedding method used for link prediction [7]. It learns low-dimensional node representations from random walks on the graph. Since these walks follow the existing network structure, node2vec may be biased by tightly connected communities. In a polarized graph, this means that pure node2vec recommendations may preserve or strengthen separation between groups.

In this project, we aim to study whether this effect can be reduced through *reranking*. Reranking can be applied after a base model has already scored the candidate links. It does not require training a new link prediction model. Instead, it keeps the original node2vec score as a measure of connection plausibility and changes the final order of candidate links.

Reranking is a common way to balance relevance with another objective. In information retrieval, diversity-based reranking has been used to select results that are both relevant and different from already selected results [8]. In recommender systems, reranking has also been used as a post-processing step to reduce unwanted recommendation bias, while preserving recommendation accuracy [9].

This leads to a trade-off. A pure node2vec ranking is expected to prioritize structurally plausible links. A diversity-aware ranking may select links that are slightly less likely according to node2vec, but more useful for connecting different opinion groups or communities. The main goal is to test whether this trade-off can reduce polarization.

The research question we aim to answer is:

Can diversity-aware reranking of node2vec-based link predictions reduce polarization after opinion dynamics, while preserving link prediction quality?

To answer this question, we use synthetic LFR graphs with communities and opinions. The graph generation setup is inspired by the framework of Cinus et al. [10], where synthetic graphs are used to control community structure and opinion similarity. We then test whether opinion-diversity and community-diversity reranking reduce polarization compared to pure node2vec after simulating opinion changes over time.

The rest of the paper is structured as follows. Section 2 discusses the background needed to understand the paper. In Section 3, relevant related literature is presented and the research gap is pointed out. Section 4 presents our approach. Section 5 describes the experimental setup and reports on the reached results. We comment on the result and present some recommendations for future research in Section 6. Section 7 presents the final conclusion of the research. Section 8 deals with responsible research.

## 2 Background

In this section, we introduce some of the key concepts necessary to understand the following sections of the paper.

### 2.1 Online Polarization & Echo Chambers

*Polarisation* happens when individuals or communities become increasingly divided into opposing groups, with larger

differences between their opinions and less interaction or agreement across those groups. The communities become more separated, and their positions may become more extreme over time [11; 12].

In social networks, polarization manifests itself via the *echo chamber* effect. In echo chambers, users are repeatedly exposed to information that reinforces their existing beliefs, while opposing perspectives receive little attention [1].

## 2.2 Opinion Dynamics

*Opinion Dynamics* is the study of how an individual’s opinion changes over time through social interaction, and how local interactions produce collective outcomes, such as consensus or polarization [13].

In a typical opinion dynamics model, each person is represented as a node in a social network and has an associated opinion. The opinion of person  $i$  at time  $t$  can often be expressed as a numerical value, such as:

$$o_i(t) \in [0, 1]$$

At each iteration, users update their opinions according to a predefined rule based on the opinions of their neighbors.

## 2.3 Link prediction for Recommender Systems

Link prediction aims to infer which new links are likely to appear in the future in an observed graph  $G$  [6]. Traditional approaches often use structural similarity, such as common neighbors, graph distance, or other proximity-based measures.

In social network analysis, link prediction is used in several ways. For example, it can be used to predict which posts to show in a user’s feed, as well as suggesting potential connections between users [14]. This is a specific form of recommender systems, which are methods designed to identify and rank items that are likely to be relevant to a particular user.

## 2.4 Node Embeddings and node2vec

Node embedding methods learn low-dimensional vector representations of nodes. The primary objective of the embeddings is to ensure that the geometric proximity of the learned vectors reflects the structural similarity of the corresponding nodes in the original graph [15].

Uniform random walks capture the structure of local neighborhoods well, but real-world graphs typically exhibit two types of structural similarities [7]:

- Homophily: nodes that belong to the same community should be embedded closely together
- Structural Equivalence: nodes that share similar network functions (e.g. nodes that act like bridges between different communities) should have similar embeddings, regardless of their proximity or connection in the graph

To address the second objective, node2vec was introduced. The algorithm learns embeddings by performing random biased walks on the graph. At each step, the transition probability is defined as follows:

$$\pi_{vx} = \alpha_{pq}(t, x) \cdot w_{vx}$$

Here,  $t$  is the previous node,  $v$  is the current node and  $x$  is a potential next node.  $w_{vx}$  is the weight of the edge between  $v$  and  $x$  and  $\alpha_{pq}(t, x)$  is a search bias term:

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

The two hyperparameters  $p$  and  $q$  are called return and in-out parameter, respectively.  $p$  controls the likelihood of going back to the previously visited node  $t$ . A low value will keep the walk in the local neighborhood, resembling a breadth-first search.  $q$  controls the likelihood of exploring outwards nodes. A low value encourages the walk to behave similarly to a depth first search.

## 2.5 Post-hoc Reranking

Post-hoc reranking is a technique for balancing a primary relevance objective against a secondary goal. It originated in information retrieval and it was used to reorder retrieved documents, while reducing redundancy among the results [8]. A single parameter controls how much relevance is traded for the secondary goal, and the underlying model is never re-trained.

## 3 Related Work

In this section, we start by reviewing the evidence that people recommenders increase polarisation. Afterwards, we review prior attempts to reduce polarization through interventions to the links the recommenders produce. We continue by reviewing post-hoc reranking as an established technique in adjacent fields. Finally, we present another research direction that can emerge from the combination.

Previous studies have shown that people recommender systems tend to reinforce existing groups and foster echo chambers, leading to increased polarisation [5; 10].

In order to mitigate the effects, several methods aim to reduce polarization by changing the links the recommender produces. They differ in where they intervene: in the recommendation objective directly, in the similarity score, or in the embedding.

Firstly, in [16], a method is proposed that selects  $k$  new links to minimise the sum of *polarization* and *disagreement* in a social network using a greedy algorithm. They define disagreement as how much connected nodes disagree in their opinions globally across the network. In this study, polarization measures how opinions deviate from their average across the whole population. The final opinions are called *equilibrium* opinions and are computed analytically, rather than through simulating opinion dynamics.

Secondly, intervening in recommendations has also been studied at the ranking stage. In [17], a diversity-aware variant of common-neighbor link prediction is introduced. The recommendations were shown to increase the diversity of connections. However, their evaluation stops at the network structure and does not test any evolution over time.

Lastly, other methods intervene during the embedding training stage. Fairwalk [18] modifies the node2vec random

walk so that neighbors are sampled uniformly across the values of sensitive demographic attributes. The study has also demonstrated that node2vec as a people recommender is biased along these sensitive attribute. The study does not deal with political opinions and requires a complete retraining of the model.

Post-hoc reranking has been previously used in recommender systems to reduce popularity bias [9]. The method promotes less popular items, while keeping accuracy acceptable. It was also shown to outperform methods that build diversity into training.

Fairness-aware reranking has also been applied to a popular talent search platform [19] to address gender fairness, without affecting business metrics [19].

So far, literature shows that reranking is an effective post-processing tool to balance a secondary objective. However, polarization was not yet studied as an objective.

Previous work thus shows a gap when it comes to applying post-hoc reranking to a people recommender system to reduce polarisation. Our work intends to fill this gap and also measure the impact of reranked links on polarisation after opinions propagate through the network, using opinion dynamics models.

## 4 Approach

In this section, we outline our overall contribution. We start by explaining our baseline prediction model. We then continue with presenting our approach with respect to post-hoc reranking. We finish by explaining how we model opinion dynamics.

### 4.1 Link Prediction Model

We assume a social network graph  $G$ , where nodes are interpreted as users and edges as connections between them. Furthermore, each user  $u$  has an associated opinion,  $o_u \in [0, 1]$ .

Before training the prediction model, we construct a test set  $T$  from  $G$ . The set is the union of  $T_{pos}$ , made of  $M$  positive edges, and  $T_{neg}$ , with  $M$  negative edges. The positive edges are extracted from the initial graph  $G$  by removing  $M$  edges, and the negative edges are derived from sampling  $M$  random pairs of non-connected nodes. Let us define the resulting graph with the removed edges as  $G'$ .

Afterwards, we learn 64-dimensional node embeddings on graph  $G'$  using node2vec. The output will be a vector of embeddings:  $Z = (z_1, z_2, \dots, z_N)$ . Using the embeddings, we calculate the baseline score using cosine similarity [20]:

$$s(u, v) = \cos(z_u, z_v)$$

### 4.2 Diversity-Aware Re-ranking

The novelty of the project consists of two node reranking methods applied as post-processing to the original scores. Assuming  $s(u, v)$  is the baseline scoring for two nodes  $u$  and  $v$ , we introduce two reranked scoring measures: opinion diversity and community diversity.

$$s_o(u, v) = \lambda \cdot s(u, v) + (1 - \lambda) \cdot |o_u - o_v|$$

$$s_c(u, v) = \lambda \cdot s(u, v) + (1 - \lambda) \cdot 1[c_u \neq c_v]$$

Where  $o_u$  and  $c_u$  represent the opinion value and community of node  $u$ , respectively. Moreover,  $\lambda$  is a trade-off parameter. It controls how much weight we give to the objective we try to balance: accuracy and diversity.

### 4.3 Opinion Dynamics

We are interested in how the added links affect opinion dynamics over time. Therefore, we have chosen two methods that model different ways users encounter each other's views on social media. We run these models after adding the top  $M$  edge predictions to the graph  $G'$ . We then measure the polarization in the graph after they run.

**DeGroot** The first model is the DeGroot model [21], in which each node updates its opinion towards the average of its neighbors. Assuming  $o_u^{(t)}$  is the opinion value of node  $u$  at time  $t$ , and  $N(u)$  is the neighborhood of  $u$ , the opinion update is as follows:

$$o_u^{(t+1)} = o_u^{(t)} + \beta \left( \frac{1}{|N(u)|} \sum_{v \in N(u)} o_v^{(t)} - o_u^{(t)} \right)$$

$\beta$  is a parameter that controls the convergence rate: how fast the opinion of a node converges towards the mean opinion of their neighborhood.

We consider this model to resemble exposure to a social media feed. At each step, the user is exposed to an aggregate of content from all of their connections. We make the assumption that this content will represent the opinion of each respective neighbor.

**BCM** The second model is called the Bounded Confidence Model (BCM) [22]. In this model at each step, nodes interact in random pairwise encounters  $(u, v)$  and only influence each other if their opinions are sufficiently similar:

$$o_u^{(t+1)} = \begin{cases} o_u^{(t)} + \gamma(o_v^{(t)} - o_u^{(t)}) & \text{if } |o_u^{(t)} - o_v^{(t)}| < \epsilon \\ o_u^{(t)} & \text{otherwise} \end{cases}$$

Here,  $\epsilon$  is the *confidence threshold*, that controls how similar opinions should be to be considered, and  $\gamma$  is the *persuasion strength*, controlling how strong the influence of a considered neighbour.

We have chosen this method to model direct interaction between users. For example, this could happen in a direct private message, or as an exchange in a comment thread. In this case, users might convince each other if their opinions are already close enough.

## 5 Experimental Setup and Results

In the following section, we start by describing our data generation algorithm. We continue by discussing and motivating the configuration parameters used. Afterwards, we present our results.

## 5.1 Random Network Generation

Distinct social networks can have different graph structures. This results from the specific ways people interact with each other on those networks. Research has shown that even comparing the same set of users from one social network to another can result in different network properties [23].

Moreover, sampling a representative graph from large social networks remains an open question. Previous work has shown that methods for subsampling from large graphs have some issues in preserving the network structure [24]. In social networks specifically, it was shown that sampling through crawling results in bias toward higher degree nodes. Furthermore, algorithms proposed to fix this have issues on fragmented social graphs [25].

Considering the above mentions, as well as the scope of the project and limited computational resources, we look towards synthetic graph generation.

We base our graph generation work on an extension of the LFR model [26]. The model generates graphs where both node degree and community sizes follow power-law distributions. This was first observed in real social networks: there are celebrities with many followers, and standard users with relatively few connections [27]. The model takes several input parameters. First,  $N$ , the number of nodes in the graph. Second,  $\tau_1$  and  $\tau_2$ , the exponents of the power law distribution of node degrees and community sizes, respectively. Thirdly, the mixing parameter  $\mu$ , which describes how segregated the different communities are. Moreover, it also takes the average node degree and minimum community sizes as parameters.

The extension was introduced by [10]. The main difference is the addition of opinion values for each node. Therefore, it adds a new parameter, the initial homophily  $\eta$ , that describes how close the opinion of a node is from the opinion of its neighbourhood. The assumption of the model is that a node can have the opinion of its underlying community, or an opinion of its own.

The resulting opinions are created by running a Bernoulli trial with probability  $\eta$ . If the outcome is positive, then the node assumes the opinion of its assigned community  $k$ ,  $O_k \sim U(0, 1)$ . Otherwise, it will form an opinion of its own, taken from the same distribution.

The output consists of a graph  $G$ , and the node opinion vector  $O \in [0, 1]^N$ . Using this method, we can control both the structure and opinions in the networks we generate by tuning  $\mu$  and  $\eta$ . Figure 1 shows graphs corresponding to the extreme values for each parameter. When the mixing parameter  $\mu$  is small and  $\eta$  is high, the result is a very polarized graph. In contrast, as  $\eta$  decreases, there will be more opinion variation within communities. Moreover, as  $\mu$  increases, the community structure becomes less visible.

## 5.2 Configuration

We generate multiple graphs using our described generation approach. For the node degree distribution, we have chosen the exponent  $\tau_1 = 2.75$ . This is motivated by the range observed in real world networks, between 2 and 3. The distribution of the community sizes has an exponent  $\tau_2 = 1.1$ , which we keep from the paper that introduced the generation model

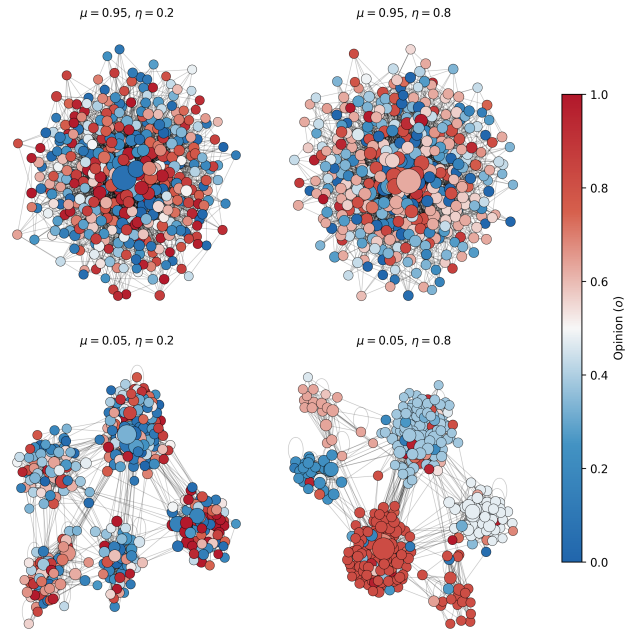


Figure 1: Examples of graphs and user opinions generated with our model with different  $(\eta, \mu)$  combinations.

[10]. We also keep graph size  $N = 400$  nodes, the average degree 12 and the minimum community size of 20.

We want to capture multiple graph topologies in our analysis. Therefore, our experiment runs a grid sweep over the two graph parameters: modularity  $\mu \in \{0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95\}$  and homophily  $\eta \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$ . For each  $(\mu, \eta)$  combination, we generate 5 graphs with distinct random seeds. In total, we generate 350 graphs.

Moreover, we set  $M$  to 30% of the total number of edges. We consider this to be sufficiently high to make a difference in the structure of the graph, and low enough to not be computationally expensive. For each node2vec run, we use a walk length of 30 and 100 walks per node. Default and in-out parameters are  $p = q = 1.0$ .

For our reranking methods, we experiment with multiple values of the  $\lambda$  trade-off parameter:  $\lambda \in \{0.8, 0.5, 0.2, 0.0\}$ . For the DeGroot model, we set the convergence rate  $\beta = 0.1$  and run 10 synchronous rounds. For the Bounded Confidence Model, we set the confidence threshold  $\varepsilon = 0.2$  and persuasion strength  $\gamma = 0.2$ . We have chosen lower convergence rate and persuasion strength in order to capture the fact that a user’s opinion is not necessarily very easily influenced.

## 5.3 Evaluation Methods & Results

Our experiments aim to answer the following three research questions, subdivisions of our main question:

**RQ1.** How do diversity-aware reranking methods affect prediction quality?

In order to answer this research question, we measure link prediction accuracy using the generated test set  $T$ . An accurate prediction model will contain more edges from  $T_{pos}$  than

from  $T_{neg}$  in the top  $M$  predictions. Assuming  $R_M$  is the set of the top  $M$  edge predictions, we calculate Hits@M:

$$\text{Hits@M} = \frac{1}{|R_m|} \sum_{e \in R_m} 1[e \in T_{\text{pos}}]$$

Figure 2 shows average *Hits@M* for each reranking method and trade-off parameter. The results indicate that reranking does lead to a reduction in prediction quality. Opinion diversity exhibits a lower impact than community diversity. Community diversity also shows considerably higher standard deviation for  $\lambda = 0.5$  and  $\lambda = 0.2$ . However, for both reranking methods, the reduction is greatly influenced by the trade-off parameter and can be controlled.

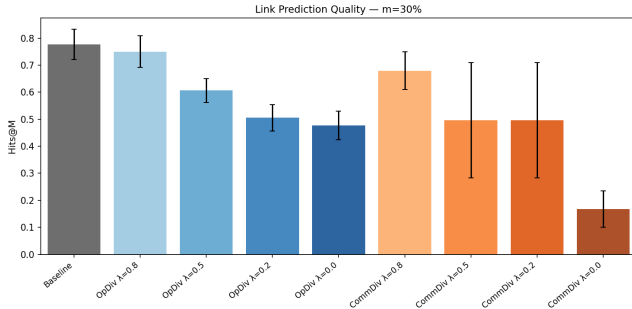


Figure 2: Average Hits@M

**RQ2.** How do the diversity-aware reranking methods affect polarization change after running opinion dynamics models?

We have chosen to measure polarisation as the mean absolute deviation from the neutral opinion, 0.5:

$$P(o) = \frac{1}{N} \sum_{u=1}^N |o_u - 0.5|$$

To evaluate the impact of reranking, we compute the *change in polarisation* for all the scoring methods after 10 rounds of each opinion dynamics model:

$$\Delta P = P(o_r^{(T)}) - P(o_b^{(T)})$$

Where  $o_b^{(T)}$  and  $o_r^{(T)}$  are the user opinion at time  $T$  after adding the baseline and the reranking method  $r$ , respectively.

In figures 3 and 4, we present the average  $\Delta P$  for each reranking method. For DeGroot, opinion diversity has a higher impact on polarization than community diversity. Moreover, the decrease in polarization is directly correlated to the decrease in accuracy. Also, varying  $\lambda$  between 0.5 and 0 does not result in the same difference in polarization change as varying it from 0.8 to 0.5. In absolute value, the overall changes are kept between 0.003 and 0.025.

For BCM, polarization slightly goes up when using opinion diversity reranking. In contrast, community diversity results in lower polarization for all  $\lambda$  values.

**RQ3.** How does the structure of the graph influence the polarization change?

In order to answer this research question, we make use of the same polarization metrics as for RQ2. Taking space constraints into account, we include the heatmaps from running DeGroot with a  $\lambda$  of 0.5 for both our reranking methods in figures 5 and 6. The top left quadrant will contain the graphs with well-separated communities and opinion consensus within each community, as previously shown in Figure 1. The impact of the reranking methods is higher for this type of networks.

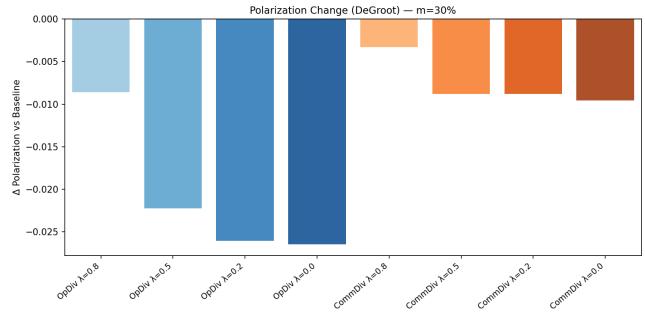


Figure 3: Average Polarization Change (DeGroot)

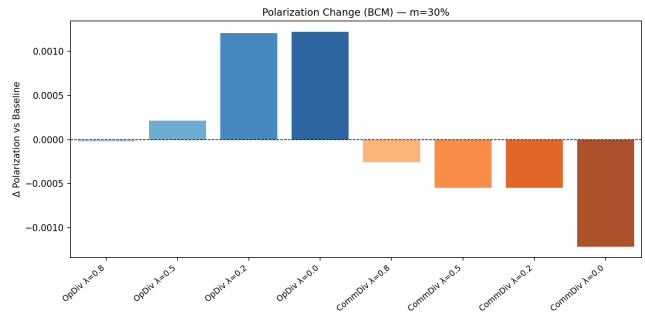


Figure 4: Average Polarization Change (BCM)

## 6 Discussion & Future Work

In this section we will discuss the results and evaluation methods. Afterwards, we will propose future research directions that can overcome current limitations and generalize the results.

Firstly, the way we model opinion dynamics has an influence on the results. Specifically, opinion diversity is more effective than community diversity for DeGroot. However, under the BCM model, it slightly increases polarization. We can attribute this to the underlying equation of the model. The model uses random pairwise encounters between neighbors at each step. Nodes are only affected by neighbors whose opinion is close enough to their own. Users with different opinions will be re-ranked higher. If the opinions of the recommended users are too different, they will effectively be ignored. Thus, the only contributing nodes will be the ones with

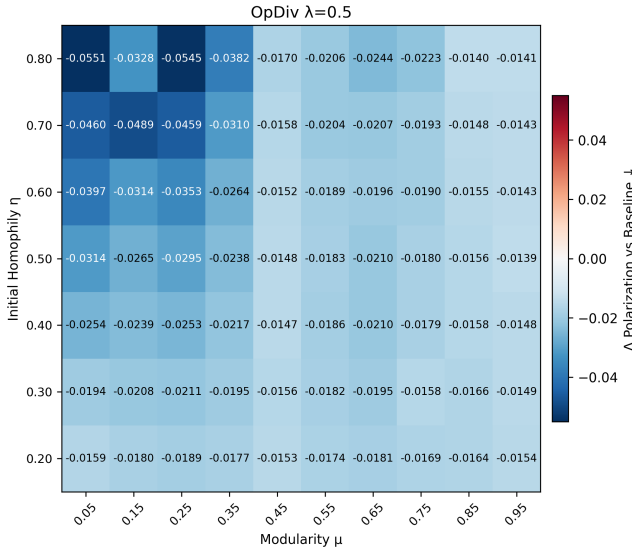


Figure 5: Heatmap of DeGroot polarization change for every  $(\eta, \mu)$  combination after applying opinion diversity with  $\lambda = 0.5$

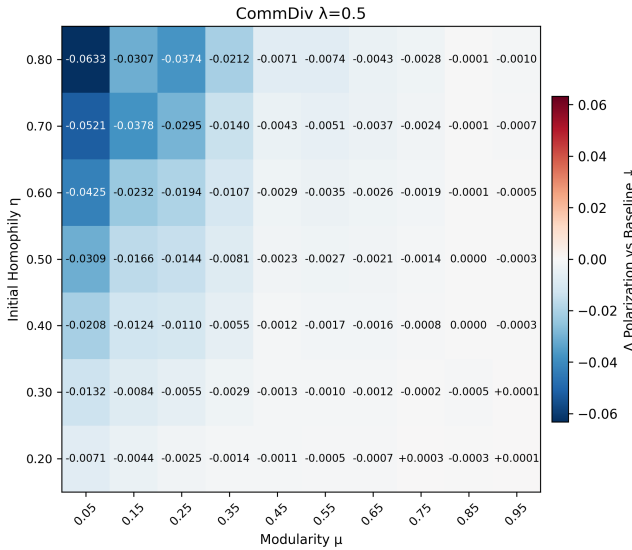


Figure 6: Heatmap of DeGroot polarization change for every  $(\eta, \mu)$  combination after applying community diversity with  $\lambda = 0.5$

close enough opinions. In contrast with this, community-diversity reranking decreases polarization under both opinion dynamics models.

Therefore, in order to propose effective reranking methods, further research is needed to better understand the way opinions are spread in social networks.

Secondly, our current accuracy model is highly simplistic and it does not model the probability for a recommendation to be accepted. Future work could include deploying the link recommendation algorithm on a social network and comparing the acceptance rates.

Thirdly, our method is currently limited to node2vec-based recommendations. However, post-hoc reranking could be applied, in theory, to any recommender algorithm that is based on scoring. Future work could include analyzing the effectiveness of the diversity-aware reranking on different underlying recommendation models.

## 7 Conclusion

To conclude, this paper proposed a new approach to apply post-hoc reranking to node2vec-based link prediction. This was used to balance two objectives, accuracy and diversity, in people recommender systems. We introduced two reranking methods: opinion diversity and community diversity.

This was done to reduce polarization in online social networks. To measure polarization change, we have applied two opinion dynamics models: DeGroot and Bounded Confidence Model (BCM).

The reranking methods have shown negligible differences in reducing polarization. However, we found that the form of reranking has different effects based on the opinion model used. The effect of opinion diversity was opposite for the two models: decreasing polarization for DeGroot, but increasing it for BCM. For community diversity, polarization decreased under both models. Moreover, we found accuracy to be directly negatively correlated with giving more weight to the diversity trade-off parameter.

There are three main limitations of the work. Firstly, the accuracy model is simplistic and does not take into account the probability of a recommendation becoming a real connection. Secondly, we only studied the effect on node2vec-based link predictions. Lastly, although the algorithm was tested on multiple graph configurations, this is still not a perfect proxy for a real social network.

## 8 Responsible Research

This section covers responsible research questions: how we applied research integrity methods to our paper, and discloses the use of generative AI.

### 8.1 Research Integrity

In this subsection, we will discuss some integrity principles outlined in the Code of Conduct for Research Integrity [28]. For *transparency*, all of the parameters and steps in the process have been clearly documented and motivated. With the experimental setup description, the results of the paper can be fully reproduced. Furthermore, we have made the code used

for computing the results public via a GitHub repository<sup>1</sup>. We respect the principle of *honesty* by reporting the results and limitations accurately, without omitting relevant findings. We follow the principle of *responsibility* by considering the possible societal effects of recommender systems and studying ways to reduce the harmful effects.

## 8.2 Use of Generative AI

I used Generative AI tools throughout the research process as a support tool. For coding, Windsurf Pro tier was used as the coding agent tool, on the "Adaptive" setting. I decided to use an agentic coding tool because of the integration with the local file system and of its task planning capabilities. The "Adaptive" setting was used to automatically select the model that optimizes the credit to performance ratio for each task. Planning would happen with models like Claude Opus 4.6. Once a plan is created, the agent will automatically switch to a lighter model, such as Claude Haiku 4.5 for the implementation. A demonstrative example of how I have used the coding agent throughout the project can be found in Appendix A.1.

In addition, I used the web version of ChatGPT free tier with GPT-5.5. I considered the model to be an appropriate tool as it is considered to be one of the most capable models on the market and includes a free tier. The model was used to support tasks such as understanding related work, summarizing parts of the research paper and formulating LaTeX equations. In addition, for the previous two drafts of the paper, generative AI was used for initial structure and writing assistance. However, those sections are not included anymore. Hence, the final submission does not contain sections for which AI was used for writing support, other than formulating LaTeX equations. A non-exhaustive set of example interactions is provided in Appendix A.2.

I did not use Generative AI to invent experimental results, fabricate citations, or make final research decisions. The final methodological choices, implementation decisions, interpretation of results, and written claims remain my responsibility.

## A Use of Generative AI

### A.1 Example Use of Windsurf

This is a representative example of how I have approached code generation for specific parts of the project. This is not meant to be exhaustive, but to provide a reference. After reviewing the plan and gathering clarifying questions from the agent, I proceed to review the code and possibly suggest improvements/fixes when I see them.

**User** We need to implement the graph generation approach used in this paper: {paper link} in our project. Make a plan for me to review and ask clarifying questions.

#### Windsurf

Based on my research of the paper, here's a plan for implementing their graph generation approach along with clarifying questions: {redacted proposed initial plan}

**Clarifying Questions** Before proceeding, I need to understand your specific requirements:

- **Scope:** Do you need the full simulation framework (graph generation + opinion dynamics + recommenders) or just the graph generation component?
- **Language/Framework:** Any preference for Python (NetworkX, igraph) or another language?
- **Scale:** What graph sizes are you targeting? (The paper uses 100-1000 nodes typically)
- **Purpose:** Is this for research replication, extending the model, or educational purposes?
- **Opinion Representation:** Should opinions be: Continuous values (e.g., -1 to 1)? Discrete categories? Binary?
- **Community Structure:** Should the model explicitly assign nodes to communities first, or should communities emerge from homophily?

Please answer these so I can tailor the implementation precisely to your needs.

**User** We just need the graph generation in Python. The graph should be 400 nodes. This is for research, and opinions should be continuous values from 0 to 1. You should stick with the original LFR generation for community assignment.

**Windsurf** {..generate to-do list, code, etc}

### A.2 Example Use of ChatGPT

This section provides examples of how I used the web version of ChatGPT with GPT-5.5 during the research process. The examples are not exhaustive, but illustrate the types of prompts used. The outputs were not used directly without checking. For factual claims, references, and paper summaries, I validated the answers by comparing them with the original papers. For mathematical notation and LaTeX equations, I checked that the formulas matched the intended implementation and compiled correctly in Overleaf. For writing and structure suggestions, I selected and edited the parts that were relevant to my own project.

**LaTeX formulation:** How can I write this score in LaTeX:  $\lambda * n2v + (1-\lambda) * (1 \text{ if cross-community else } 0)$ ?

- **Answer:**

You can write it using an **indicator function**:

```
\[
\operatorname{score}(u, v)
=
\lambda \cdot \operatorname{n2v}(u, v)
+
(1-\lambda)
\mathbb{I}[\!\left[c_u \neq c_v\right],
\]
```

where  $(\mathbb{I}[c_u \neq c_v])$  equals (1) when (u) and (v) belong to different communities, and (0) otherwise.

Alternatively, make the condition explicit:

```
\[
\operatorname{score}(u, v)
```

<sup>1</sup><https://github.com/tmihaila/polarization>

```

=
\lambda \, \operatorname{n2v}(u,v)
+
(1-\lambda)
\begin{cases}
1, & \text{if } c_u \neq c_v, \\
0, & \text{otherwise.}
\end{cases}
\end{cases}
\]

```

- Validation: The relevant part of the answer is put into LaTeX, I verify the formatting, and make any adjustments, e.g. modifying *score* to *s*.

**Reference search:** In which paper was the LFR model first introduced?

- Answer: The LFR benchmark was introduced by Lancichinetti, Fortunato, and Radicchi in 2008. It generates synthetic networks in which node degrees and community sizes follow power-law distributions, making it more representative of real-world networks than earlier benchmarks. The original paper is available at <https://doi.org/10.1103/PhysRevE.78.046110>.
- Verification: Navigate to the paper link and read it.

**Concept Clarification:** What is the difference between the SBM and LFR graph generation models?

- Answer: The Stochastic Block Model (SBM) and the LFR benchmark both generate networks with community structure, but they make different assumptions.

In a standard SBM, nodes are assigned to communities, and edges are created according to fixed probabilities for pairs of communities. Nodes within the same block are therefore statistically similar, and their degrees are usually relatively homogeneous.

The LFR benchmark is designed to resemble real-world networks more closely. It generates power-law degree distributions and power-law community-size distributions, allowing large variation in both node degrees and community sizes. Its mixing parameter ( $\mu$ ) controls the fraction of each node's edges that connect outside its community.

Therefore, SBM is simpler and easier to analyze, while LFR produces more heterogeneous and realistic benchmark networks. A degree-corrected SBM can model heterogeneous degrees, but LFR still explicitly combines power-law degrees, power-law community sizes, and tunable community mixing.

- Validation: Consult the original papers to validate the information is accurate.

**Writing support:** Generative AI was used in draft v1 for writing parts of the Related Work section, that also overlap with part 2.1 and 2.2 of draft v2. However, the subsections are no longer part of the final submission.

**Section Structure:** Some sections in the draft v1 and v2 were structured using Generative AI. However, the sections are not part of the final submission anymore.

**Paper understanding:** I mostly use the three-pass approach when reading new papers. Generative AI was used after the first pass, to summarize the main ideas before doing the second pass. This helped to create a mindmap of the overall paper and approaches. Moreover, it was only used for the experimental pipeline, and not for the results.

- Example: Make a summary of how Cinus et al. set up their experiment, including the graph generation process, recommendation strategies, opinion dynamics models, and polarization metrics.

- Answer: Cinus et al. use an extended LFR benchmark to generate directed social networks with community structure and node opinions. The parameter ( $\mu$ ) controls how many links connect different communities, while ( $\eta$ ) controls how strongly users' opinions are aligned with the opinion assigned to their community. They evaluate four people-recommendation strategies: Directed Jaccard, Personalized PageRank, SALSA, and an opinion-based recommender that favors users with similar opinions.

During the simulation, accepted recommendations create new links, while random outgoing links are removed to keep the network density approximately constant. The authors then model opinion evolution using either the Bounded Confidence Model, in which sufficiently similar users move their opinions closer together, or the Bala–Goyal epistemological model, in which users update beliefs from shared evidence. Echo-chamber formation is measured with the Neighbor Correlation Index, which captures the similarity between users and their neighbors, while global polarization is measured using Random Walk Controversy. The impact of each recommender is evaluated by comparing the final polarization and echo-chamber scores with a corresponding simulation without recommendations.

- Validation: I validated the results of the summary by doing the second pass of the paper, keeping in mind possible differences. The results were not intended to act as a proxy for the paper, but only for creating a mindmap that would facilitate understanding.

## References

- [1] Matteo Cinelli, Gianmarco De Francisci Morales, Alessandro Galeazzi, Walter Quattrociocchi, and Michele Starnini. The echo chamber effect on social media. *Proceedings of the National Academy of Sciences*, 118(9):e2023301118, 2021.
- [2] Rafael Evangelista and Fernanda Bruno. Whatsapp and political instability in brazil: Targeted messages and political radicalisation. *Internet Policy Review*, 8(4), 2019.
- [3] Jon Green, William Hobbs, Stefan McCabe, and David Lazer. Online engagement with 2020 election misinformation and turnout in the 2021 georgia runoff election. *Proceedings of the National Academy of Sciences*, 119(34):e2115900119, 2022.

- [4] Eduard-Claudiu Gross and Tanjev Schultz. Social media dynamics in the 2024–2025 romanian presidential election campaign: TikTok’s disruptive role and the enduring importance of journalism and legacy media. *Journalism Research*, 8(2):188–206, 2025.
- [5] Fernando P. Santos, Yphtach Lelkes, and Simon A. Levin. Link recommendation algorithms and dynamics of polarization in online social networks. *Proceedings of the National Academy of Sciences*, 118(50):e2102141118, 2021.
- [6] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.
- [7] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 855–864, 2016.
- [8] Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.
- [9] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher. Managing popularity bias in recommender systems with personalized re-ranking. In *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference*, 2019.
- [10] Federico Cinus, Marco Minici, Corrado Monti, and Francesco Bonchi. The effect of people recommenders on echo chambers and polarization. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):90–101, 2022.
- [11] Aaron Bramson, Patrick Grim, Daniel J. Singer, Steven Fisher, William J. Berger, Graham Sack, and Carissa Flocken. Disambiguation of social polarization concepts and measures. *The Journal of Mathematical Sociology*, 40(2):80–111, 2016.
- [12] Kieran Mephram, Andr’as V’or’os, and Christoph Stadtfeld. Network polarization: The study of political attitudes and social ties as dynamic multilevel networks. *Network Science*, 13:e7, 2025.
- [13] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2):591–646, 2009.
- [14] Zhepeng Li, Xiao Fang, and Olivia R. Liu Sheng. A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions. *ACM Transactions on Management Information Systems*, 9(1):1–26, 2018.
- [15] Sami Abu-El-Haija, Bryan Perozzi, Rami Al-Rfou, and Alex Alemi. Watch your step: Learning node embeddings via graph attention. *arXiv preprint arXiv:1710.09599*, 2017.
- [16] Lijing Zhu, Hunter Z. Brooks, and Mason A. Porter. Minimizing polarization and disagreement in social networks via link recommendation. In *Advances in Neural Information Processing Systems*, volume 34, 2021.
- [17] Robert Warton, Cupcake Volny, and Kevin S. Xu. Counteracting filter bubbles with homophily-aware link recommendations. In *Social, Cultural, and Behavioral Modeling*, volume 13558 of *Lecture Notes in Computer Science*, pages 155–164. Springer, 2022.
- [18] Tahleen A. Rahman, Bartlomiej Surma, Michael Backes, and Yang Zhang. Fairwalk: Towards fair graph embedding. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pages 3289–3295, 2019.
- [19] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. Fairness-aware ranking in search & recommendation systems with application to linkedin talent search. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD ’19)*, pages 2221–2231, 2019.
- [20] Gerard Salton, Anita Wong, and Chung-Shu Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
- [21] Morris H. DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [22] Guillaume Deffuant, David Neau, Frederic Amblard, and Gerard Weisbuch. Mixing beliefs among interacting agents. *Advances in Complex Systems*, 3(01n04):87–98, 2000.
- [23] Simón Peña-Fernández, Ainara Larrondo-Ureta, and Jordi Morales-i Gras. Straddling two platforms: From Twitter to Mastodon, an analysis of the evolution of an unfinished social media migration. *Social Sciences*, 14(7):402, 2025.
- [24] Harish Sethu and Xiaoyu Chu. A new algorithm for extracting a small representative subgraph from a very large graph, 2012.
- [25] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. Walking in Facebook: A case study of unbiased sampling of OSNs. In *Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM)*, pages 2498–2506, 2010.
- [26] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110, 2008.
- [27] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement, IMC ’07*, pages 29–42, New York, NY, USA, 2007. ACM.

- [28] KNAW, NFU, NWO, TO2-Federation, Netherlands Association of Universities of Applied Sciences, and VSNU. Netherlands code of conduct for research integrity. Technical report, The Hague, 2018.