Bias Reduction in Social Networks through Agent-Based Simulations

Nathan Bartley$^{1[0000-0002-6450-5476]}$, Keith Burghardt$^{1[0000-0003-1164-9545]}$, and Kristina Lerman$^{1[0000-0002-5071-0575]}$

Information Sciences Institute, Marina Del Rey CA, 90292 USA

Abstract. Online social networks use recommender systems to suggest relevant information to their users. Studying how these systems expose people to information at scale is difficult to do as one cannot assume each user is subject to the same feed condition and building evaluation infrastructure is costly. We show that a simple agent-based model where users have fixed preferences affords us the ability to compare different personalization algorithms in their ability to skew users' perception of their network. Importantly, we show that a simple greedy algorithm that chooses users based on network properties reduces such perception biases comparable to a random feed. This underscores the influence network structure has in determining the effectiveness of recommendation systems and offers a tool for mitigating perception biases through algorithmic feed construction.

Keywords: Social Networks · Agent-based Models · Recommendations.

Recommender systems are pervasive in online social media platforms and they span various functions including social link recommendations (e.g., "Who to Follow"), ad targeting (i.e., users are recommended for ad providers), geolocated news (e.g., Trending Topics), and content recommendation more broadly. While these recommender systems afford users the ability to sift through incredible amounts of information online, they have become the objects of study and critique for their plausible yet not well understood role in amplifying information [17, 13].

User preferences have recently been explored in agent-based models on YouTube regarding their primary video recommender system: however the comparison of different systems and feeds has not been fully explored, especially in online social networks that present personalized feeds like X and Facebook [18, 6]. Recent works study the differences between the black-box algorithmically personalized and reverse chronological feeds on both X and Facebook, however these works focus more on what content is being shown to users rather than who the users are being exposed to, which can effect how information spreads [10, 11].

In this paper we make the following contributions:

1. We present scaleable agent-based model simulations with $173,000$ nodes.
2. We compare perception biases generated from baselines, deep-learning approaches, and a novel greedy algorithm for personalizing news feeds.

3. We demonstrate that this greedy algorithm creates less biased feeds and makes feeds that are comparable in utility to the other tested models.

1 Related Work

This work can largely be categorized under social network simulations, recommender system analysis, and recommender system auditing, as we establish a framework that can be used to plug in and analyze how different implementations of recommender systems in personalized feeds work.

1.1 Social Network Simulations

Social network multi-agent simulations for recommender system analysis have been largely focused on information diffusion and predicting user behavior in different environmental circumstances. In Muric et al. [16] the authors utilize Twitter, Reddit and GitHub data to understand information spread, especially across different online platforms. This study did not explicitly focus on comparing different personalized feeds but rather focused on predicting information cascades in online environments.

Murdock et al. [15] simulated user and moderation behavior on Reddit to assess the impact of community-based networks and the unique moderation structure the platform has for belief diffusion.

Ribeiro, Veselovsky, and West [18] utilized an agent-based model to address the paradox that content-based recommender systems face: seemingly favoring extreme content, these systems do not seem to be the primary driver of what users consume. When incorporating a measure of user utility in choosing which content to engage with, results suggest users will not necessarily engage with suggested content if it is of low utility. Our work differs in that we compare different recommender systems and how they would behave considering the same user behavior patterns.

Donkers et al. [6] studied both epistemic and ideological echo chambers in social media and the effects of diversifying recommendations on discussions. They used knowledge graph embeddings to make diverse recommendations in a retweet network to depolarize discussions between users with varying propensities for accepting new information (i.e., retweeting something a peer posted). In our work we simplify the models and compare their effect on perception.

1.2 Recommender System Audits

In general recommender system audits have been focused on content-based recommender systems as they are the most straightforward to analyze. In particular, most recent work has been focused on YouTube and the role the video recommendation engine might have in spreading misinformation and radicalization. These works, like Tomlein et al. [20] and Hussein et al. [13] used agent-based sock puppets to simulate user behavior directly on the platform to measure the response in terms of personalization and prevalence of misinformation.

Both Spinelli and Crovella [19] and Ribeiro et al. [17] investigated YouTube's video recommender system and how it might contribute to gradual user radicalization. The latter study in particular analyzed user interactions and comments, focusing on migrations of users between communities. It is relevant because it is important to trace users in their experience on the platform to really understand how recommender systems and users interact. However, these works being focused on YouTube precludes studying how social signals play a role in the user-recommender system loop: the same information may be perceived positively by a user if it was presented as being "liked" by a friend, but negatively by that user if it was presented by someone they do not know or actively dislike. This makes such content-based studies about recommendations less relevant for platforms like X and Facebook.

1.3 Personalized News Feeds

Most of the studies around the effects of personalized news feeds, defined as the rank ordered list of information a user would receive when logging onto a platform like Facebook or X/Twitter, are focused around user satisfaction, impact on information diffusion, and echo chambers.

Two papers addressing perception and user satisfaction are Eslami et al. [8] and Eslami et al. [9], where tools were developed to study how users perceive the Facebook news feed algorithm at that time, i.e., they studied the impact of algorithmic awareness on user satisfaction and perception of their networks.

Bakshy et al. [3] from Facebook addressed information diffusion via the personalized News Feed by measuring user exposure to ideologically diverse posts. In this they considered how user preferences and algorithmic influence play a role in content consumption. They focused on the dynamics of information diffusion in Facebook's network at the time and did not emphasize the effect of different personalized news feeds.

Among several Meta studies published in 2023, two are of particular interest in understanding impacts of personalization in news feeds. Guess et al. [11] investigated the Facebook and Instagram feed algorithms and their potential impact on user attitude and election behavior in the 2020 presidential election. Gonzalez-Bailon et al. [10] studied the interaction of algorithmic curation and user social amplification by studying the spread of political URLs on Facebook, showing a segregated experience between liberal and conservative users. These papers are relevant to studying social network based recommendation systems, however they are focused on political content diffusion and political-behavior related outcomes for real users: the present study is concerned with comparing different types of personalized feeds and how they may impact *who* users are exposed to. This exposure is important as perception of the prevalence of traits in a network can potentially impact beliefs and behaviors related to those traits.

2 Model

2.1 Framework

We use the Repast framework to run models over a cluster of compute nodes [5]. This framework has previously been used in other areas like simulating bike-sharing systems in cities, and complex biological systems requiring heterogeneous multicellular organisms [14, 2]. A key factor facilitating the use of Repast is that in our simplified network, users will only be exposed to the tweets that their friends (and friends-of-friends) generate, thus allowing us to partition the network and run them in different processes.

2.2 User Behavior

In this work we trace the behavior of approximately 173,000 users sharing 1.5 million edges, and examine the experience of 5,599 central users as they follow the below sequence of events for each time tick t:

1. Activate user i with a uniform probability (0.083)
2. If activated, user i then produces a certain number of tweets; we sample a lognormal distribution ($\mu = 0.0, \sigma = 1.0$) to choose how many tweets the user produces in that time period
3. Add created tweets to the content pool
4. "Backend" model serves tweets to user i if appropriate
5. User i likes tweet from another user with same label at fixed probability (0.20), with probability (0.05) otherwise

Our model is a discrete event-based model, where for each time tick we "step" the individual nodes and then update model information before proceeding to the next time tick.

There are two key components for the model: the "backend" model that serves users tweets and the network of users. While each user is connected via the network, they interact with other users on the network through the model by sending tweets to the model and getting tweets from their friends through the model. This way we can vary how tweets are served to users.

Two events happen during each simulation:

- At tick $t = 12$ we increase the ρ_{kx} to approximately 0.25
- At tick $t = 24$ we increase the ρ_{kx} to approximately 0.50 and reset the edges seen to reflect a full 24 hours passing

This is to assess three dynamics of the system: 1) what happens when we have a cold start; 2) what happens when we change how the trait is distributed across the users by swapping labels between them; and 3) what happens when we "forget" most information from the previous day.

We define ρ_{kx} as:

$$\rho_{kx} = \frac{P(x=1)}{\sigma_x \sigma_k} [\langle k \rangle_{x=1} - \langle k \rangle] \qquad (1)$$

where $\langle k \rangle_{x=1}$ is the average in-degree of the "active" $x = 1$ nodes, and $\langle k \rangle$ is the average in-degree of all nodes considered.

We structure the network based on data from Alipourfard et al. [1] who gathered a complete follower network for $5,599$ users, as well as tweets and retweets for those users and everyone they followed to generate a dataset with 4M users from May - September 2014. We use this data to guide our model; we downsample the nodes for the sake of simulation runtime.

2.3 Model Parameters

We treat each simulation time step as a single hour, for a total of 36 timesteps. Per a blog-post from X, users spend an average of 32 minutes per day on the platform, which we implement in an activation probability: each user has a 0.083 probability of "logging in" per hour to give an expected value of approximately two sessions per 24 hours [22].

Each simulation has every user subjected to the same personalized news feed:

1. **Random.** All candidate tweets are randomly sorted and the first n tweets are served to the user.
2. **Reverse Chronological.** All candidate tweets are sorted in reverse chronological order, and the first n tweets are served to the user.
3. **Neural Collaborative Filtering.** We implement a simple version of the Neural Collaborative Filtering (NCF) model to demonstrate how a deep learning model more broadly can be used for recommender systems in this context. We train the model on the 5,599 core users, where each tweet liked is the "item" being trained on. We keep the model straightforward and only use the superficial user level information [12].
4. **Wide & Deep.** We implement a simple version of the Wide & Deep model described initially by researchers at Google to demonstrate how a recommender system used in production in other contexts might behave in this scenario [4]. Similar to the NCF model we use user-level features for training.
5. **Minimize $\rho_{\mathbf{kx}}$.** We implement a greedy strategy for choosing which edges to observe for each user. We use eqn. 1 and sort the tweets seen by a user at every tick t by how much that edge would contribute to the difference between the mean "active" in-degree $\langle k \rangle_{x=1}$ and mean in-degree $\langle k \rangle$, opting for the edge that would minimize the difference.

We chose these personalized news feed algorithms to analyze different implementations of news feeds: there is a public release of the X (formerly Twitter) recommendation "algorithm", however as it relies on multiple models and active services it would be unreliable to use the code as-is in a simulation (especially as production parameters to the models have since changed) [23]. Instead we aim to show simple baseline models, two deep-learning models, NCF and Wide & Deep, and our greedy strategy for minimizing exposure bias (MinimizeRho).

To assess perception of networks, we randomly assign each user in the network a binary trait $X \in \{0, 1\}$ such that the total prevalence of the trait in the network is static. We run each simulation under $P(X = 1) \in \{0.05, 0.15, 0.50\}$ to assess the impact of the prevalence of the trait on the behavior of the system. Each user, based on the value of the trait that they are assigned, also behaves in a biased manner towards the tweets that they observe: users with $x = 1$ will like tweets from users with $x = 1$ with probability 0.20 and otherwise like tweets with probability 0.05. Likewise for users with $x = 0$. Both numbers were chosen to elicit, in expectation, at least one like from each active user per tick.

Each simulation similarly has every user using the same length feed, i.e., they only observe (and potentially engage) a fixed number of tweets for each timestep in the simulation. We tested lengths of 30, 50, 100.

2.4 Bias & Performance Metrics

We use two metrics to study the bias of this perception:

1. Local Bias (B_{local}) $B_{\text{local}} = E[q_f(\text{X})] - E[f(\text{X})]$
2. Gini Coefficient (G) $G = \frac{\sum_{i=1}^{n}(2i-n-1)x_i}{n\sum_{i=1}^{n}x_i}$

We define B_{local} as the average frequency of the attribute among a node's immediate network: $E[q_f(\text{X})] = \bar{d} * E[f(U)A(V)|(U,V) \sim \text{Uniform}(E)]$; $E[f(X)]$ is the global frequency of the node attribute f (here, 0.05, 0.15, 0.50); $f(U)$ the attribute value f of node U; $\bar{d}$ represents the expected in-degree of the network. $A(V)$ represents the “attention“ node V pays to any node in their network. B_{local} should vary from [-1,1], and Gini should vary from [0,1], where 0 is equal and 1 is unequal. This plays the role of an overall view of the skew users will experience in their personalized feeds.

Gini coefficient in this context represents the skew in the number of unique friends (or friends-of-friends) users are exposed to in their feeds: x_i represents the number of times that friend (or friend-of-friend) was observed.

We use several other measurements to study the simulation and verify results (some not reported here):

1. **Precision@30.** mean $\left(\frac{|\text{tweets liked in first 30 positions}|}{30}\right)$
2. **Number of edges seen.** Total unique edges seen up until that time tick, including friends-of-friends.
3. **Mean number of likes friends' tweets receive.** We take the sum total likes each friend receives from core users and take the mean over all friends.

3 Results

Across models in Fig. 1 we observe relatively stable bias for the network until the reset at $t = 24$. Interestingly, we observe the Random and MinimizeRho conditions having consistently low measures (in absolute value) of B_{local} (Local Bias). Similarly, the NCF and WideDeep conditions are correlated to one another, showing lower values than the other models until the reset at $t = 24$.

In Fig. 2 we observe stability in the behavior of the Gini Coefficient, where Random and MinimizeRho conditions remain the lowest in terms of Gini, suggesting a more even distribution of attention across friends. The Chronological condition tends to have the next lowest Gini coefficient. The NCF and WideDeep conditions tend to have the highest Gini Coefficient, suggesting a narrower focus on sets of friends observed by users. This remains consistent until the $t = 24$ reset where the MinimizeRho condition jumps up in Gini.

For validity checks of the simulations we present the results in Figs. 3 and 4. Fig. 3 shows that the core users all behave similarly under different conditions in terms of the number of tweets that they like cumulatively over the course of the simulation. The longer feed length simulations tend to have higher mean likes than the shorter ones. We observe similar behavior in the mean number of likes each friend receives, with longer feeds having higher mean tweets cumulatively over time. Finally we observe the unique edges observed by the core users in Fig. 4, where the Random and MinimizeRho conditions maximize the total observed edges by allowing more edges to be observed.

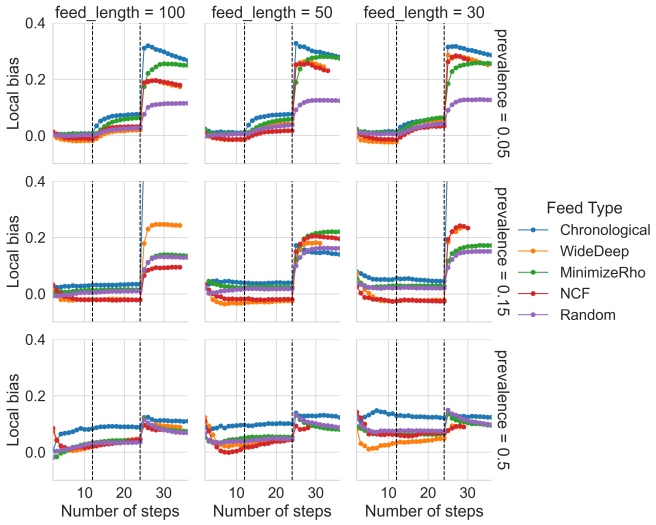

Fig. 1. Local Bias (B_{local}). Graph depicts the difference between the expected local fraction of friends who have $x = 1$ and the true global prevalence of the trait $P(X = 1)$. Positive implies over-representation, negative implies under-representation.

The precision figure in Fig. 5 demonstrates that all model conditions improve over time, with different levels of precision drop after the $t = 24$ reset.

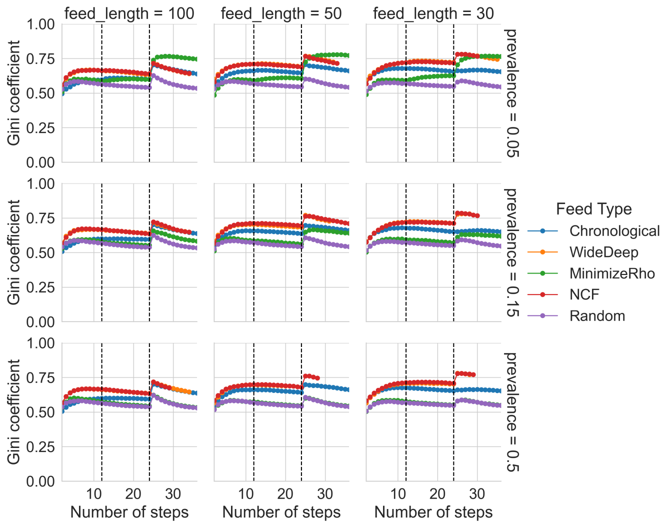

Fig. 2. Gini Coefficient. Graph depicts the distribution of times each friend (or friend-of-friend) was observed by a core user. 1 implies inequality, 0 implies equality.

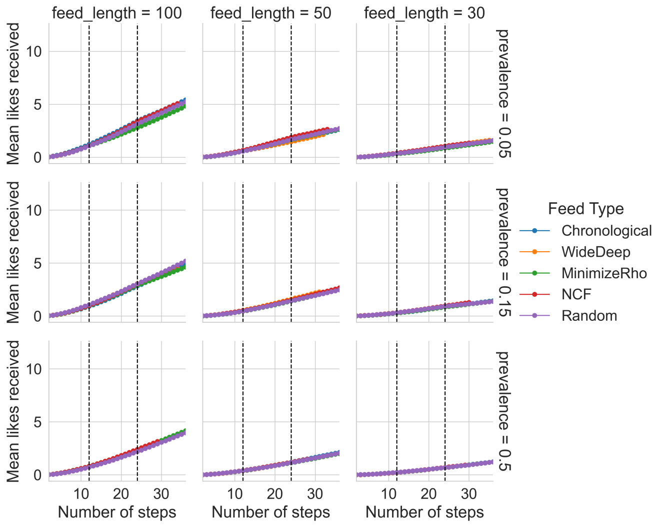

Fig. 3. Mean number of likes each friend receives . Graph depicts the mean number of likes each friend receives over time.

4 Discussion

The information presented in Fig. 4 suggests that the Random and MinimizeRho conditions demonstrate the maximum growth in number of unique edges

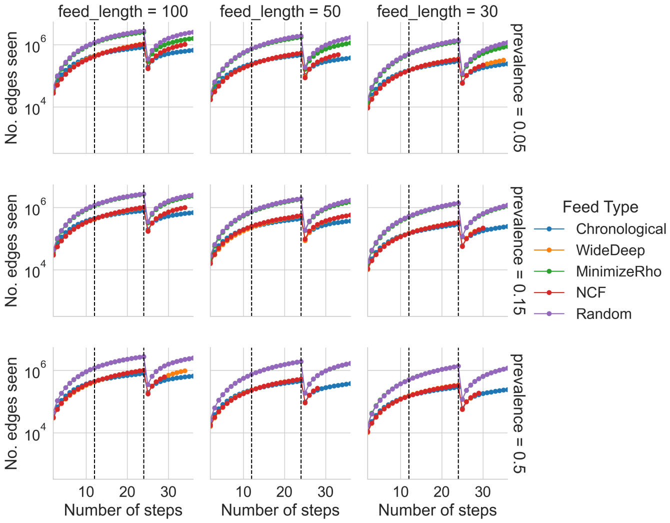

Fig. 4. Log number of friends seen. Graph depicts the log total number of unique friends (and friends-of-friends) seen through tick t. Connections seen are reset at $t = 24$.

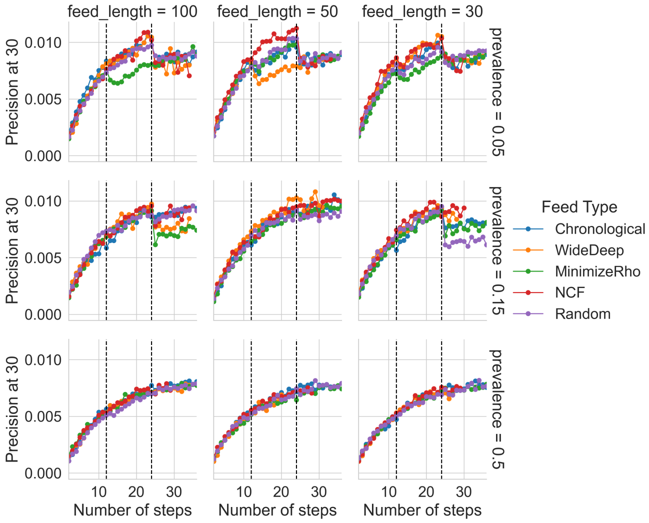

Fig. 5. Precision@30. Graph depicts the total fraction of liked tweets in the first 30 positions in the feed

observed over time, increasing diversity in users who are observed. As this might be due to spurious changes in user behavior, we measure the mean number of likes each friend receives in Fig. 3. Here we observe that the longer feeds provides more likes for each friend, which follows given the increased number of "chances" each user would get to like a friend's tweet. Given each model has users that behave similarly, we then use the perception measures in Figs. 1 and 2 to determine differences between models. Two patterns show up: the Random and MinimizeRho conditions correlate closely and lower in absolute value in B_{local} and G, however the NCF and WideDeep models behave tightly as well, with lower values in B_{local} but higher values in G. This suggests that the deep learning models are more tightly focused on certain sets of users (as corroborated in the number of edges seen in Fig. 4). If the Random model can be considered less biased than the others by minimizing the absolute value, then the MinimizeRho model seems to be most closely related to it in behavior over time.

Curiously, each model trends upwards in performance in Precision@30 in Fig. 5. We would expect the Random model to have minimal slope as it does not depend on the user feedback, however, as the number of likes is stored cumulatively we would anticipate a growing number over time. The differences show themselves across different prevalences: $P(X = 1) = 0.05$ demonstrates the NCF model is reliably higher than the other models in both measures of precision (Precision@10 not reported here), but the difference in models disappears as the prevalence of the trait tends towards $P(X = 1) = 0.50$. The drop in performance of the MinimizeRho model at feed length 100, $P(X = 1) = 0.05$ may be explained by potentially different assignments of the trait X after $t = 12$ as the WideDeep model shows peculiar behavior at feed length 50 for the same prevalence.

5 Limitations & Future Work

One clear limitation is the duration we ran the simulations. Another limitation is that we do not try more recommender systems and longer personalized feeds. There are advanced recommender systems that could be useful to analyze, like MV-DNN [7] or the stated X/Twitter system [23], however adapting the models is difficult as many require access to richer information about the users and content than we have built here (or in a production system access to other microservices or models the system depends on).

Using Large Language Models as agents interacting in the framework would be interesting future work, considering Tornberg et al. [21] and their preliminary work in this space. This could facilitate the use of complicated recommender systems, as other ways of generating richer information would potentially make the ABM too contrived to be useful.

We would also like to integrate more metrics into the analysis, as there may be confounding factors present in our simulations (e.g., B_{local} and G may converge but some other measure might be periodic). Similarly we would like to extend this analysis to more than binary user labels as labels can change over time and are often more complicated than simple binaries. In an effort for reproducibility we release the simulation scripts.[1]

[1] https://anonymous.4open.science/r/cuddly-octo-couscous-9111/

6 Conclusion

In this work we describe an agent-based model and framework for studying the effects of different personalized news feed algorithms in online social networks in how they expose users to their networks. The model and framework is extensible and given the underlying MPI usage of the underlying Repast library very scalable contingent upon having access to an MPI-enabled computing resource. We find that while deep learning methods are useful and tend to minimize perception bias in terms of our binary label, they focus on a narrower set of users. We find that a simple greedy algorithm based on network properties keeps diversity in attention and minimizes our measure of local perception bias B_{local}.

These findings are important for designing recommender systems in online social networks that are robust: these systems mediate the information and connections between people and we should be able to understand what happens as people interact with these systems.

7 Ethical statement

We generally believe that agent-based models are an appropriate method for studying these systems in a way that preserves user privacy and dignity in this area of research. We do have concerns however that the more we understand recommender systems, the more of an attack vector we open up for malicious actors to manipulate these systems. Considering the recent (as of this writing) layoffs at major social media platforms, and a changing focus in Trust and Safety, this could pose a problem for the spread of harassment and misinformation.

References

1. Alipourfard, N., Nettasinghe, B., Abeliuk, A., Krishnamurthy, V., Lerman, K.: Friendship paradox biases perceptions in directed networks. Nature communications **11**(1), 707 (2020)
2. Bae, J.W., Lee, C.H., Lee, J.W., Choi, S.H.: A data-driven agent-based simulation of the public bicycle-sharing system in sejong city. Simulation Modelling Practice and Theory **130**, 102861 (2024)
3. Bakshy, E., Messing, S., Adamic, L.A.: Exposure to ideologically diverse news and opinion on facebook. Science **348**(6239), 1130–1132 (2015)
4. Cheng, H.T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 7–10 (2016)
5. Collier, N., North, M.: Repast hpc: A platform for large-scale agent-based modeling. Large-Scale Computing pp. 81–109 (2011)
6. Donkers, T., Ziegler, J.: The dual echo chamber: Modeling social media polarization for interventional recommending. In: Proceedings of the 15th ACM Conference on Recommender Systems. pp. 12–22 (2021)

7. Elkahky, A.M., Song, Y., He, X.: A multi-view deep learning approach for cross domain user modeling in recommendation systems. In: Proceedings of the 24th international conference on world wide web. pp. 278–288 (2015)
8. Eslami, M., Aleyasen, A., Karahalios, K., Hamilton, K., Sandvig, C.: Feedvis: A path for exploring news feed curation algorithms. In: Proceedings of the 18th acm conference companion on computer supported cooperative work & social computing. pp. 65–68 (2015)
9. Eslami, M., Karahalios, K., Sandvig, C., Vaccaro, K., Rickman, A., Hamilton, K., Kirlik, A.: First i" like" it, then i hide it: Folk theories of social feeds. In: Proceedings of the 2016 cHI conference on human factors in computing systems. pp. 2371–2382 (2016)
10. González-Bailón, S., Lazer, D., Barberá, P., Zhang, M., Allcott, H., Brown, T., Crespo-Tenorio, A., Freelon, D., Gentzkow, M., Guess, A.M., et al.: Asymmetric ideological segregation in exposure to political news on facebook. Science **381**(6656), 392–398 (2023)
11. Guess, A.M., Malhotra, N., Pan, J., Barberá, P., Allcott, H., Brown, T., Crespo-Tenorio, A., Dimmery, D., Freelon, D., Gentzkow, M., et al.: How do social media feed algorithms affect attitudes and behavior in an election campaign? Science **381**(6656), 398–404 (2023)
12. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of the 26th international conference on world wide web. pp. 173–182 (2017)
13. Hussein, E., Juneja, P., Mitra, T.: Measuring misinformation in video search platforms: An audit study on youtube. Proceedings of the ACM on Human-Computer Interaction **4**(CSCW1), 1–27 (2020)
14. Montagud, A., Ponce-de Leon, M., Valencia, A.: Systems biology at the giga-scale: Large multiscale models of complex, heterogeneous multicellular systems. Current Opinion in Systems Biology **28**, 100385 (2021)
15. Murdock, I., Carley, K.M., Yagan, O.: An agent-based model of reddit interactions and moderation (2023)
16. Murić, G., Tregubov, A., Blythe, J., Abeliuk, A., Choudhary, D., Lerman, K., Ferrara, E.: Large-scale agent-based simulations of online social networks. Autonomous Agents and Multi-Agent Systems **36**(2), 38 (2022)
17. Ribeiro, M.H., Ottoni, R., West, R., Almeida, V.A., Meira Jr, W.: Auditing radicalization pathways on youtube. In: Proceedings of the 2020 conference on fairness, accountability, and transparency. pp. 131–141 (2020)
18. Ribeiro, M.H., Veselovsky, V., West, R.: The amplification paradox in recommender systems. In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 17, pp. 1138–1142 (2023)
19. Spinelli, L., Crovella, M.: How youtube leads privacy-seeking users away from reliable information. In: Adjunct publication of the 28th ACM conference on user modeling, adaptation and personalization. pp. 244–251 (2020)
20. Tomlein, M., Pecher, B., Simko, J., Srba, I., Moro, R., Stefancova, E., Kompan, M., Hrckova, A., Podrouzek, J., Bielikova, M.: An audit of misinformation filter bubbles on youtube: Bubble bursting and recent behavior changes. In: Proceedings of the 15th ACM Conference on Recommender Systems. pp. 1–11 (2021)
21. Törnberg, P., Valeeva, D., Uitermark, J., Bail, C.: Simulating social media using large language models to evaluate alternative news feed algorithms. arXiv preprint arXiv:2310.05984 (2023)
22. X: One year in. https://blog.twitter.com/en$_us/topics/company/2023/one-year-in(2023), accessed : 2024-04-06$

23. X: Twitter github. https://github.com/twitter/the-algorithm-ml (2023), accessed: 2024-04-06