Simulating News Recommendation Ecosystems for Insights and Implications

Guangping Zhang, Dongsheng Li, *Senior Member, IEEE*, Hansu Gu, *Member, IEEE*, Tun Lu, *Member, IEEE*, Li Shang, *Member, IEEE*, and Ning Gu, *Member, IEEE*

***Abstract*—Studying the evolution of online news communities is essential for improving the effectiveness of news recommender systems. Traditionally, this has been done through empirical research based on static data analysis. While this approach has yielded valuable insights for optimizing recommender system designs, it is limited by the lack of appropriate datasets and open platforms for controlled social experiments. This gap in the existing literature hinders a comprehensive understanding of the impact of recommender systems on the evolutionary process and its underlying mechanisms. As a result, suboptimal system designs may be developed that could negatively affect long-term utilities. In this work, we propose SimuLine, a simulation platform to dissect the evolution of news recommendation ecosystems and present a detailed analysis of the evolutionary process and underlying mechanisms. SimuLine first constructs a latent space well reflecting the human behaviors and then simulates the news recommendation ecosystem via agent-based modeling. Based on extensive simulation experiments and the comprehensive analysis framework consisting of quantitative metrics, visualization, and textual explanations, we analyze the characteristics of each evolutionary phase from the perspective of life-cycle theory and propose a relationship graph illustrating the key factors and affecting mechanisms. Furthermore, we explore the impacts of recommender system designing strategies, including the utilization of cold-start news, breaking news, and promotion, on the evolutionary process, which sheds new light on the design of recommender systems.**

***Index Terms*—Agent-based modeling, life cycle, news recommendation, online community, simulation, social impact.**

Manuscript received 22 May 2023; revised 13 December 2023; accepted 9 March 2024. This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62172106 and Grant 61932007. *(Corresponding author: Tun Lu.)*

Guangping Zhang, Li Shang, and Ning Gu are with the School of Computer Science and Shanghai Key Laboratory of Data Science, Fudan University, Shanghai 201203, P. R. China (e-mail: gpzhang20@fudan.edu.cn; lishang@fudan.edu.cn; ninggu@fudan.edu.cn).

Dongsheng Li is with Microsoft Research Asia, Shanghai 200241, P. R. China, and also with the School of Computer Science, Fudan University, Shanghai 201203, P. R. China (e-mail: dongshengli@fudan.edu.cn).

Hansu Gu resides in Seattle, WA 98109 USA (e-mail: hansug@acm.org).

Tun Lu is with the School of Computer Science, Shanghai Key Laboratory of Data Science, Fudan Institute on Aging, MOE Laboratory for National Development and Intelligent Governance, and Shanghai Institute of Intelligent Electronics & Systems, Fudan University, Shanghai 201203, P. R. China (e-mail: lutun@fudan.edu.cn).

Digital Object Identifier 10.1109/TCSS.2024.3381329

I. Introduction

DUE to the proliferation of social media, people are increasingly relying on online news communities to publish and acquire news. There are millions of news posted to online news communities by content creators and read by a large number of users with the distribution of recommender systems [1], [2]. Online news communities are continuously and dynamically evolving with the production and consumption of news content [3], [4], [5]. Like other online communities, online news communities evolve in line with the famous life-cycle theory [6], i.e., they would go through the phases of "start-up"–"growth"–"maturity"–"decline" in turn. Through the lens of life-cycle theory, extensive works have investigated the evolving patterns of online communities and given suggestions for operations at each stage [7], [8]. However, the impact of recommender systems, one of the most important technical infrastructures, on the lifecycle of online news communities is yet unclear. The life-cycle theory is critical for analyzing the long-term utility of recommender systems and suggesting optimal designs [9].

In the field of online community analysis, classic research methods, such as quantitative analysis based on static datasets, have provided numerous insights for optimizing platform design. However, these methods are constrained by issues such as data unavailability and the lack of open platforms for conducting social experiments, which limit the study of the dynamic evolution of online communities.

Therefore, this article will focus on the following three research questions and try to answer them via simulation experiments: 1) what are the characteristics of each phase of the evolutionary lifecycle of recommender system-driven online news communities (news recommendation ecosystems or NREs for short)?; 2) what are the key factors driving the evolution of NREs and how do these factors affect the evolutionary process?; and 3) how can we achieve better long-term multistakeholder utilities and avoid communities from falling into "decline" through the design strategy of recommender systems?

Some attempts have been made to study the social impact of recommender systems through simulations [10], [11]. These works seek to explain phenomena, such as filter bubble [12], [13], [14], popularity bias [15], [16], and content quality [17]. However, as these works focus on some specific issues, they ignore the following limitations when designing their simulator, resulting in the difficulty in modeling the evolutionary process of NREs: 1) the exposure bias of the original datasets is not handled properly. The synthetic data generation for the simulators needs to be based on real-world datasets to ensure authenticity and reliability. As the original datasets are collected from users already exposed to algorithmic recommendations, the exposure bias is introduced unavoidably, resulting in troubles such as distribution distortion and algorithm confounding [18]. 2) The latent space could not well reflect the human behaviors due to excessive idealization or limited explainability.[1] Some works regard each dimension of the latent space as a disentangled attribute and sample the latent vectors from prespecified distributions [19], [20]. This approach is over-idealistic and ignores the correlation between different features as well as their different importance. The other works learn the latent vectors from historical interactions using matrix factorization algorithms [21], [22], [23]. However, the learned latent space is abstractive and less explainable, restricting the understanding of the evolutionary process. 3) Existing frameworks are not authentic enough to reflect real-world scenarios. On one hand, existing frameworks confuse the concepts of recommendation algorithms and recommender systems. In addition to algorithmic recommendations, recommender systems need to address the cold-start issues, and strategic recommendations such as utilizing trending news and promotions should also be considered. On the other hand, the existing frameworks ignore some key factors in user behavior modeling, such as the distinction between preconsumption and postconsumption behaviors and content quality modeling [24]. In this article, we first propose SimuLine, an advanced simulation platform for dissecting the evolution of NREs. SimuLine first performs synthetic data generation based on real-world datasets. To address the inherent exposure bias of the original datasets, SimuLine introduces the inverse propensity score (IPS) for bias elimination [25]. To build a latent space close to the human decision process, we introduce the pretrained language models (PLMs) [26], [27], which are trained on large-scale corpus and proved to exhibit certain rules of human cognition [28], [29]. Then, we simulate the interaction behavior of users, content creators, and recommendation systems through agent-based modeling. Based on the extensive analysis framework, this article attempts to answer the above three research questions with life-cycle analysis, key factors and affecting mechanisms analysis, and recommender system designing strategies experiments. Our major contributions are summarized as follows.

1) We propose SimuLine, a novel news recommendation ecosystem simulation platform, supporting effective and reliable simulation of users, content creators, and recommender systems. Utilizing the PLMs and IPS, SimuLine operates in a realistic and explainable latent space, which is consistent with human behaviors and can get rid of the inherent exposure bias of the original datasets.
2) SimuLine is the first work to dissect the life cycle of the news recommendation ecosystem via simulation. We attempt to understand the characteristics of each phase and propose a relationship graph illustrating the key factors and affecting mechanisms, shedding new light on designing recommender systems responsible for long-term utilities.
3) The simulation platform and simulation experiments will be opensourced,[2] further contributing to the community for understanding and analyzing the news recommendation ecosystems in the future.

II. Related Work

Considering the high cost of online experiments and the low flexibility of dataset-based offline experiments, simulators have been widely leveraged in building recommendation models and understanding recommender systems. Earlier simulators are designed for generating synthetic data to overcome the limited availability of appropriate datasets [30], [31], [32]. These works learn the distributions of the incomplete original datasets and then apply data generating schemes for specific scenarios, such as context-awareness [30], streaming [31], and privacy preservation [32]. More recent simulators could be divided into two main categories according to their purposes. The first category contains simulators built for reinforcement learning tasks [11], [33], [34], [35], [36], [37], [38], [39], [40], [41], which are designed for optimizing the long-term user utilities in multiround interactions. Regarding the recommender system as an agent, these works construct user simulators, which could continuously respond to the recommendations, as the environment.

For example, CF-SFL [41] proposes a user simulator comprised of a reward estimator and a feedback generator, designed to facilitate the training of recommendation algorithms based on a synthetic feedback loop. MINDSim [11] constructs a user latent space for user feature sampling using generative adversarial networks (GANs), and employs an encoder–decoder architecture to learn user click behaviors and interest changes.

Besides, some methods pay additional attention to content provider modeling [36], visit pattern modeling [39], debiasing [38], etc., in order to make the simulated users more realistic. Most of these approaches are implemented with probabilistic programming to support model optimization, which demands the probability of the interaction process tractable. However, this requirement also introduces limitations on simulation scale and agent flexibility.

Additionally, these simulators model user behavior using black-box deep learning models, which limits the analysis of the driving factors behind user behaviors.

The second category contains simulators designed for understanding the social impact of recommender systems in long-term interactions. Compared to the simulators designed for reinforcement learning, these simulators loosen the requirement for probabilistic tracking and pay more attention to building complex and realistic simulation systems. These works first propose well-designed white-box simulation frameworks, reproduce the concerned phenomena, and then explain the phenomena using the simulating data. Following this methodology, the existing literature analyses the role of recommender systems in filter bubble [13], echo chamber [12], [13], popularity bias

[1] In this article, we use "latent space" to refer to the real user interest space modeled by the simulators, and "embedding space" to refer to the user interest learned by the recommendation algorithm within the simulation system.

[2] https://github.com/aSeriousCoder/SimuLine.

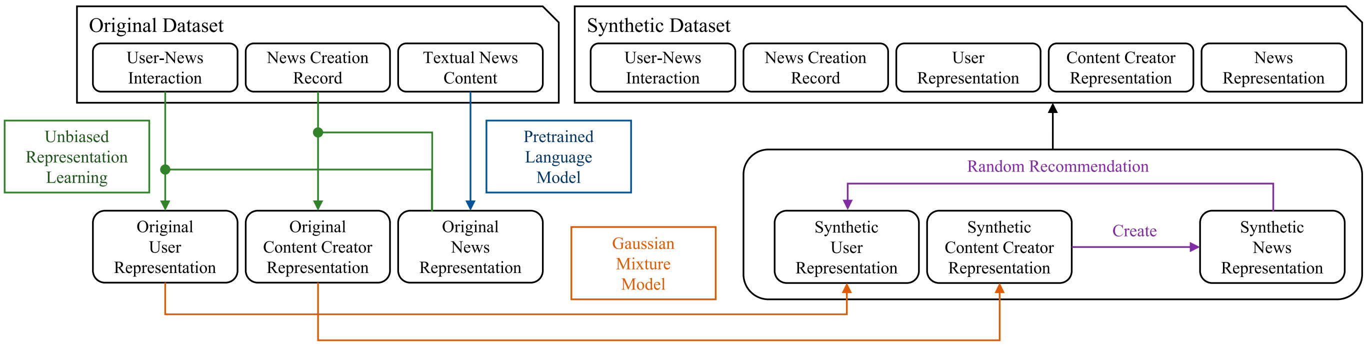


Fig. 1. Procedure of synthetic data generation. This module transforms the original dataset into a synthetic dataset, where it resolves the inherent exposure bias and generates the latent space representations of users, content creators, and news.

[22], content quality [17], feedback loops [23], user homogenization [14], [18], [20], algorithmic recommendation coverage [21], and exploration–exploitation tradeoff [19].

For example, Yao et al. [22] developed four types of choice models and two feedback models of users, discussing the impact of recommender systems on the popularity bias for different types of users. RecLab [19] simulates recommendation interactions based on six dynamic models of user interest and discusses the consistency between offline metrics and online performance, as well as the impact of exploration strategies on long-term recommendation effectiveness.

SimuLine further extends this line of work via improved synthetic data generation and simulation designs, enabling more comprehensive analysis.

Similarly, we define the action strategies of user agents based on white-box models such as preference drift models and utility-expectation models, to support in-depth analysis of the long-term evolutionary dynamics of the recommendation ecosystem. At the same time, we also introduce advanced techniques such as PLM and IPS into user agent modeling, achieving more realistic and accurate interaction simulations.

With the advancement of generative large language models (LLMs) such as ChatGPT, researchers have begun exploring the utilization of LLMs to construct agents for simulating the dynamic evolution of communities [42]. Recent works, including RecAgent [43] and Agent4Rec [44], have applied this new user modeling paradigm to recommendation simulations, achieving promising results. Compared to vectorized simulation approaches, these LLM-based simulations offer enhanced interpretability. However, the high computational demands ofbreak generative LLMs limit their efficiency, posing challenges for large-scale simulations. For instance, both RecAgent and Agent4Rec, constrained by OpenAI's API rate limits, can only support simulations of up to 1000 users. In contrast, our approach leverages discriminant PLM for efficient vectorized simulations in the latent space, making our tool a viable option for large-scale simulations of news recommendation ecosystems.

III. The Simulation Platform

This section introduces SimuLine, a simulation platform to dissect the evolution of news recommendation ecosystem. SimuLine consists of two key modules: 1) *Synthetic data generation module*, in which SimuLine builds the latent space, deals with the inherent exposure bias of the original dataset and initializes the simulation; 2) *Agent-based simulation module*, in which SimuLine builds the agents of users, content creators, and recommender system, and simulates the recommendation feedback loops.

A. Synthetic Data Generation

As the public datasets cannot meet all the specific requirements, especially the ground-truth user preference and the corresponding news features that are essential in simulating user decision process, we first generate synthetic data to initialize the simulation. More specifically, we learn a latent space based on the textual news contents and original interactions from the dataset. Then, we sample from this latent space to generate synthetic users and their interactions. As such, we can simulate the evolution from scratch and control the scale of the simulating platform by hyperparameters. In addition, since all users evolved in the simulation are synthetic, there will not be privacy concerns in our platform. Fig. 1 illustrates the procedure of synthetic data generation.

1) Building the Latent Space: The latent space is designed to model the user decision process. Each vector in the latent space represents the preference of one user or the features of one news. Here, building a latent space which could accurately reflect the human behaviors is the primary challenge. Recent success of the pretrained language models (PLMs) [26], [27] provides us with a new solution for tackling this challenge. Pretrained on large-scale unlabeled corpus via self-supervision, the PLMs can learn to encode universal textual information, preserving key properties reflecting human cognition, e.g., word analogies could be solved with the vector arithmetic on top of word encodings [28], [29]. Thus, we propose to take the semantic space of PLMs as the latent space. Using PLMs, news could be encoded directly in the latent space. For the users and content creators, we map them into the latent space via representation learning and vector arithmetic using the historical interactions. Thereby, we can provide textual explanation for every latent vector by retrieving similar vectors, making the evolution more explainable.

In our implementation, we utilize BPEmb[3] [45], which is a collection of pretrained multilingual subword embeddings based on byte-pair encoding (BPE) and trained on Wikipedia, to encode the original news.

2) Learning Unbiased User and Content Creator Latent Representation: The historical interactions of the users are derived from online algorithmic recommendations, i.e., user behaviors have been affected by the original recommender system of the news platform, resulting in the problem of exposure bias [38], [46]. We leverage the IPS [47] to deal with the exposure bias. Formally, denoting the historical interactions between the users and the news as I^U and the PLM-encoded news latent representation matrix as H^N, the goal is to learn the latent matrix H^U of the users via unbiased Bayesian personalized ranking [25] using I^U and H^N. The optimization objective is formulated as follows:

$$\max_{H^U} \frac{1}{|\mathcal{D}_{\text{pair}}|} \sum_{(u,i,j)\in\mathcal{D}_{\text{pair}}} \omega_{u,i,j} \cdot \ln \sigma\left(S_{u,i} - S_{u,j}\right) - \lambda \| H^U \|_2^2 \tag{1}$$

where $\mathcal{D}_{\text{pair}}$ refers to the pairwise dataset extracted from the historical interactions via negative sampling. $S_{u,i} = H_u^U \cdot H_i^N$ is the matching score based on the latent representations. λ is the regularization parameter. $\omega_{u,i,j} = (I_{u,i}^U/\theta_{u,i})\big(1 - (I_{u,j}^U/\theta_{u,j})\big)$, where $\theta_{u,i}$ is the IPS debiasing the exposure probability, which is defined as follows:

$$\theta_{u,i} = P\left(I_{u,i}^U = 1 \mid \sigma\left(S_{u,i}\right) = 1\right). \tag{2}$$

As news creation is not affected by the recommendation algorithms directly, we take the average of news latent vectors to form the latent representation matrix H^C of content creators.

3) Generating Synthetic Users and Content Creators: Since the online news platforms usually come with multiple news panels, the original latent representations exhibit the distributional characteristic of a superposition of multiple Gaussian distributions. Correspondingly, we utilize the Gaussian mixture model (GMM) [48] to fit the distributions and generate synthetic users. The optimization objective is formulated as

$$\max_{\gamma,\theta} \sum_{i=1}^{N} \log\left(\sum_{t=1}^{T} \gamma_i \phi(H_i^U | \mathcal{N}(\mu_t, \boldsymbol{\Sigma}_t))\right). \tag{3}$$

$\phi(\cdot|\mathcal{N}(\mu_t, \boldsymbol{\Sigma}_t))$ is the density function of the tth multivariate Gaussian distribution. γ_i is the mixing coefficient for the ith Gaussian, which represents the probability that a randomly chosen data point comes from the ith Gaussian. N is the configurable number of users, which controls the scale of the simulation. T is the number of categories in the original dataset. We take two steps to generate synthetic user i:

1) step 1: sample the category $\hat{t} \sim$ Multinomial (γ); and
2) step 2: sample the latent vector $\hat{H}_i^U \sim \mathcal{N}(\mu_{\hat{t}}, \boldsymbol{\Sigma}_{\hat{t}})$.

Then, $\hat{H}_i^U$ is used as the real preference of synthetic user i in the simulation. The generation of synthetic content creators follows the same procedure, so that the details are omitted.

[3]We choose the off-the-shelf technique for engineering convenience, as the designing choices of different PLMs or collaborative filtering-based recommendation algorithms are orthogonal to our main contributions and would not affect the main findings.

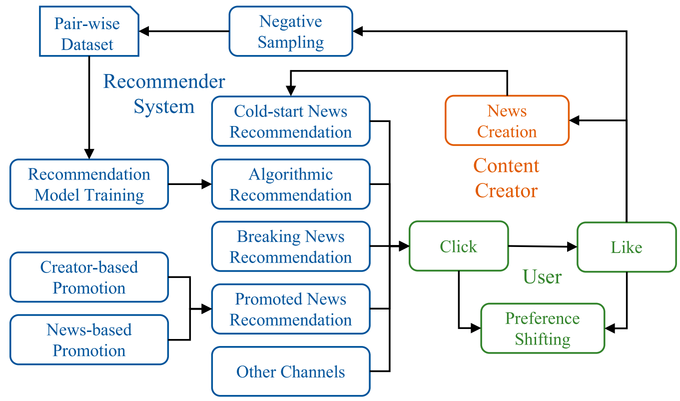


Fig. 2. Procedure of the simulation framework. The agents of content creators, the recommender system, and users take actions in turn, constituting the simulation of recommendation feedback loops.

4) Initialization for Simulation: To initialize the simulation, each synthetic content creator first samples K news from the distribution $\mathcal{N}(\hat{H}_i^C, \rho_i^C \cdot \delta)$ to build the news pool. $\rho^C \in \mathbb{R}^M$ is a vector with elements ranging from 0 to 1, representing the degree of concentration of the content creators. M is the configurable number of content creators. δ is a hyperparameter which controls the magnitude of the latent space variation. Then, the recommender system generates random recommendation lists $\mathcal{L}$, which is the default strategy when lacking historical interactions as training data. Note that the random recommendations here can also avoid the exposure bias issue during the initialization of the simulation. In the end, the synthetic users respond to the recommendations according to the action policies defined in Section III-B3.

B. Agent-Based Simulation

To model the evolution process of NRE as in detail as possible, we attempt to build a comprehensive and extensible simulation system as illustrated in Fig. 2. Next, we describe the detailed designs for the agents.

1) Agent of Content Creator: When creating news, the content creators leverage the greedy mechanism, which means that topics received more `LIKE`s in the last round would be more likely selected as the cornerstone of the future news. Formally, the probability of content creator i generating news j is defined as

$$P_{i,j}^C = \operatorname*{softmax}_{j\in N_i^C} \exp\left\{-\frac{1-\rho_i^C}{D_j^N + \varepsilon}\right\} \tag{4}$$

where N_i^C refers to the news created by content creator i. ρ^C is the concentration vector of the content creators, controlling their sensitivity to the number of received `LIKE`s. D_j^N is the number of received `LIKE`s of news j. ε is a small constant to avoid zero in the denominator. Through sampling with replacement, each content creator selects K news. Denoting the latent vector of the kth news selected by content creator i as $\hat{H}_{i,k}^{N_{\text{selected}}}$, the news is sampled from $\mathcal{N}\left(\hat{H}_{i,k}^{N_{\text{selected}}}, \rho_i^C \cdot \delta\right)$. The number of `LIKE`s is diminishing marginally correlated with the income, and the creators with higher income would create news with higher quality due to more sufficient budget. Thus, we define the quality of news as $\log(D_i^C + 1)$, where D_i^C is the number of LIKEs received by creator i in the last round [randomly drawn from Binomial distribution $\mathcal{B}(n_{\text{click}}, p_{\text{like}})$ for the initial round, where n_{click} and p_{like} are hyperparameters]. The created news will be passed to the recommender system and be recommended using cold-start recommendation strategies. Note that each news will be active for only S rounds, reflecting the timeliness of news recommendation.

2) Agent of Recommender System: The algorithmic recommendation and cold-start recommendation are the two basic elements of the recommender system agent. To provide personalized algorithmic recommendations, the recommender system agent first learns the user preference in the embedding space using recommendation algorithms, e.g., BPR [49], from historical interaction datasets. However, due to the uncertainty of user actions and the time limit for news activation, the dataset is not guaranteed to cover all the users. Hence, for those covered by the dataset, we recommend the top-scored news using the trained recommendation model. For users not in the dataset, we recommend randomly. The newly created news could not involve in the algorithmic recommendation due to lacking interaction records. The agent applies strategies such as random recommendations and news from historically liked creators, to recommend cold-start news. SimuLine also supports extensive news recommendation strategies, such as breaking news, content creator-based promotion, and topic-based promotion, via independent exposure quotas. In the end, the agent merges the recommended news from all channels to form the final recommendation list $\mathcal{L}$.

3) Agent of User: In the real-world news consumption scenarios, the users first decide whether to click on the news based on whether the title and abstract can draw their interests. After reading the news, the users will decide whether to give a LIKE according to if the content and quality of the news exceed their expectations [24]. We formulate the CLICK action as a probabilistic selection process [50], [51], and the probability of user i clicking on news j is defined as

$$P_{i,j}^U = \underset{j \in \mathcal{L}_i}{\text{softmax}} \left(\rho_i^U \cdot \frac{\hat{H}_i^U \cdot \hat{H}_j^N}{\|\hat{H}_i^U\|_2^2 \cdot \|\hat{H}_j^N\|_2^2} \right) \tag{5}$$

where ρ^U is the concentration vector of the users, controlling users' willingness to explore new content. We define the degree to which user i would like news j as the reading utility $\mathcal{U}_{i,j}$, which is determined by two factors: the content utility and the quality utility. User i will like news j if the reading utility $\mathcal{U}_{i,j}$ exceeds user i's expected threshold $\mathcal{T}_i$, which is formulated as

$$\mathcal{U}_{i,j} = \alpha_i \frac{\hat{H}_i^U \cdot \hat{H}_j^N}{\|\hat{H}_i^U\|_2^2 \cdot \|\hat{H}_j^N\|_2^2} + (1 - \alpha_i) \frac{Q_j}{\mathcal{Q}} > \mathcal{T}_i \tag{6}$$

where α_i is a hyperparameter to balance user i's preference between content utility and quality utility. Q_j refers to the quality of news j. $\mathcal{Q} = \log(K \cdot N + 1)$ is the highest quality the creators could obtain under certain simulation environment. Based on the positive or negative interactions, the users strengthen or weaken their preference on the topics, respectively. We leverage the user-drift model [21] to formulate this process as follows:

$$\hat{H}_i^U \leftarrow \hat{H}_i^U + \sum_{j \in \mathcal{L}'_i} \mathbb{I}_{i,j} \cdot \frac{\delta(\hat{H}_j^N - \hat{H}_i^U)}{\|\hat{H}_j^N - \hat{H}_i^U\|_2^2} \cdot \exp\left\{ - \frac{\rho_i^U}{\lfloor \mathcal{U}_{i,j} \rfloor_0 + \varepsilon} \right\} \tag{7}$$

where $\mathcal{L}'_i$ is the set of news clicked by user i. $\mathbb{I}_{i,j}$ is 1 if user i likes news j and -1 otherwise. $\lfloor \cdot \rfloor_0$ refers to the clipping operation with the minimal value of 0.

IV. Experiments

This section first describes the implementation details of our simulation experiments. Then, we introduce the analysis framework and present the simulation results. We attempt to answer the three research questions via life-cycle analysis, key factors and affecting mechanisms analysis, and recommender system designing strategies experiments as follows.

1) What are the characteristics of each phase of the evolutionary lifecycle of NREs? (Findings 1–6).
2) What are the factors driving the evolution of NREs and how do these factors affect the evolutionary process? (Finding 7).
3) How can we achieve better long-term multistakeholder utilities and avoid communities from falling into decline through the design strategy of recommender systems? (Finding 8).

A. Implementation Details

1) Dataset: We adopt Adressa [52], a Norwegian real-world news recommendation dataset covering one week of the whole web traffic (including 1 060 341 clicking behaviors from 133 765 users, and 14 661 news from 2611 authors) from February 2017 on the *www.adressa.no* website, as the original dataset for the consideration of data completeness.[4]

2) Synthetic Data Generation: The number of synthetic users and content creators is 10 000 and 1000, respectively. The initial user LIKE probability p_{like} is set as 0.1. We simulate 100 rounds and in each round every creator produces five news.

3) Agent-Based Modeling: The default recommender system performs algorithmic recommendations using the BPR algorithm [49] 2 and uses random recommendation for cold-start scenarios. The algorithmic model is trained using the Adam optimizer on the pairwise dataset. We take LIKEs as the positive records and sample negative records uniformly with a negative sampling ratio of $1:9$.[5] The learning rate is $1e-3$ and the batch size is 1024. The model would be trained for up

[4]Other high-quality news recommendation datasets, such as MIND, fail to meet our requirements due to the lack of author information.

[5]Although sampling negative records from clicked news or exposed news could yield better algorithmic recommendation coverage on the news, the improved coverage comes from incorporating more news as negative samples into the process of model training, of which the learned embeddings present no practical values.

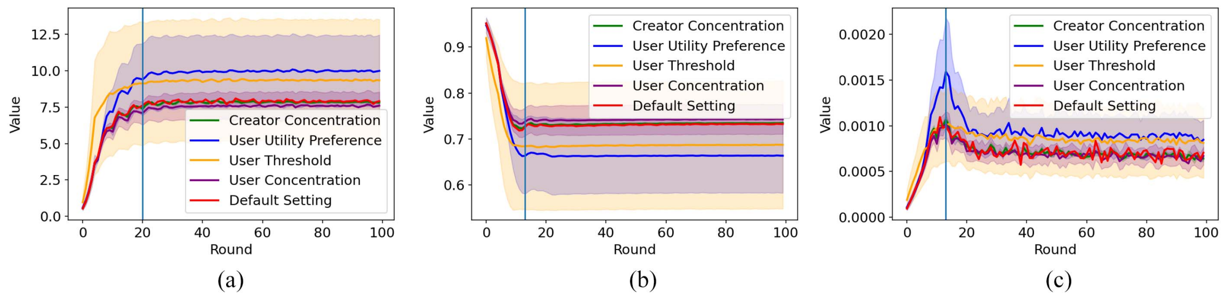


Fig. 3. User-related metrics versus rounds, including average number of (a) LIKEs, (b) Gini index of users, and (c) Jaccard index of users.

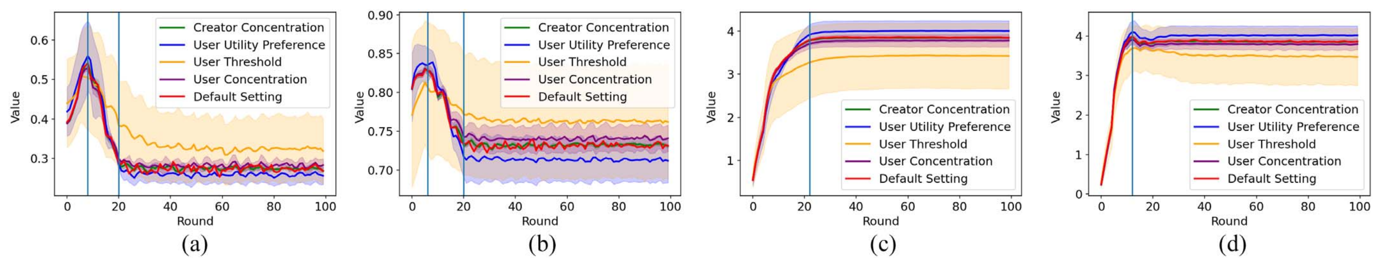


Fig. 4. Content creator-related metrics versus rounds, including Gini index of (a) creators, (b) news, (c) average quality of active news, and (d) LIKE-weighted quality of active news.

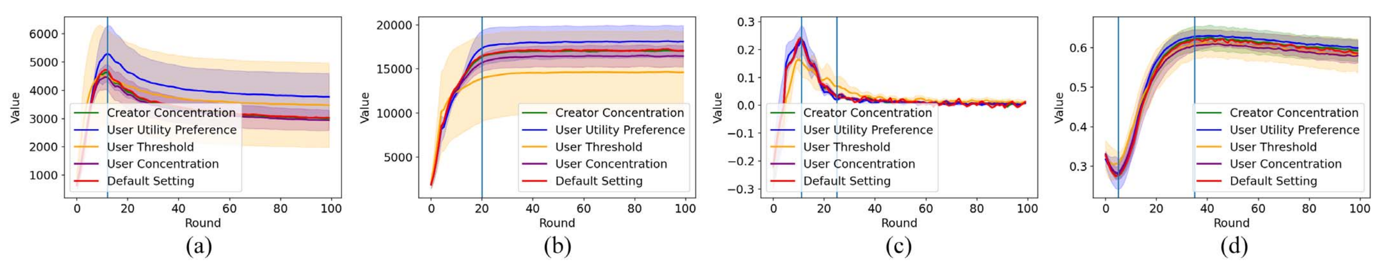


Fig. 5. Recommender system-related metrics versus rounds, including the number of (a) users, (b) news covered by algorithmic recommendation, and (c) correlation coefficient between popularity and quality, and (d) cosine similarity between users and liked news.

to 100 epochs with an early stop strategy where the threshold number of nonexceeding step is set as 3. The dataset is split as subsets for training, evaluation, and testing randomly with the ratio of 80%, 10%, and 10%, respectively. The evaluating metric is MRR@5. Each user is recommended with 100 news (80 news from algorithmic recommendation and 20 news from cold-start recommendation) in each round and the number of clicks n_{click} is set as 10. The user thresholds $\mathcal{T}$ are sampled from $\mathcal{N}(0.3, 0.1)$. The user utility preference α is sampled from $\mathcal{N}(0.5, 0.1)$. The user and creator concentration ρ^U and ρ^C are sampled from $\mathcal{N}(0.5, 0.1)$. The above four hyperparameters are clipped to be non-negative.

B. Analysis Framework and Simulation Results

We analyze the simulation results from three perspectives: quantitative metrics, latent space visualization, and latent space explanation.

1) Quantitative Metrics: To comprehensively understand the evolutionary process, we summarize and extend the quantitative metrics used in the existing literature from the following five perspectives: 1) *interaction*, including the number of LIKEs and their Gini index [15], [18]. A lower Gini index represents better fairness; 2) *coverage*, including the numbers of users and news covered by the algorithmic recommendation; 3) *quality*, including average quality of active news, LIKE-weighted average quality of active news, and the Pearson product-moment correlation coefficient between news quality and the number of LIKEs; 4) *homogenization*, including the Jaccard index [14], [18], of which a higher value represents a higher degree of overlap in news reading among the users; and 5) *latent representation*, including the cosine similarity between the latent representations of users and their liked news.

Figs. 3, 4, and 5 present the evolution of these metrics under varying environmental hyperparameters, including $\mathcal{T}$ (user threshold), α (user utility preference), ρ^U (user concentration), and ρ^C (content creator concentration). The lines indicate the mean values, and the shaded areas represent the variance. From Figs. 3, 4, and 5, we can observe that with the tenth and twentieth rounds as the approximate boundaries, all the metrics under

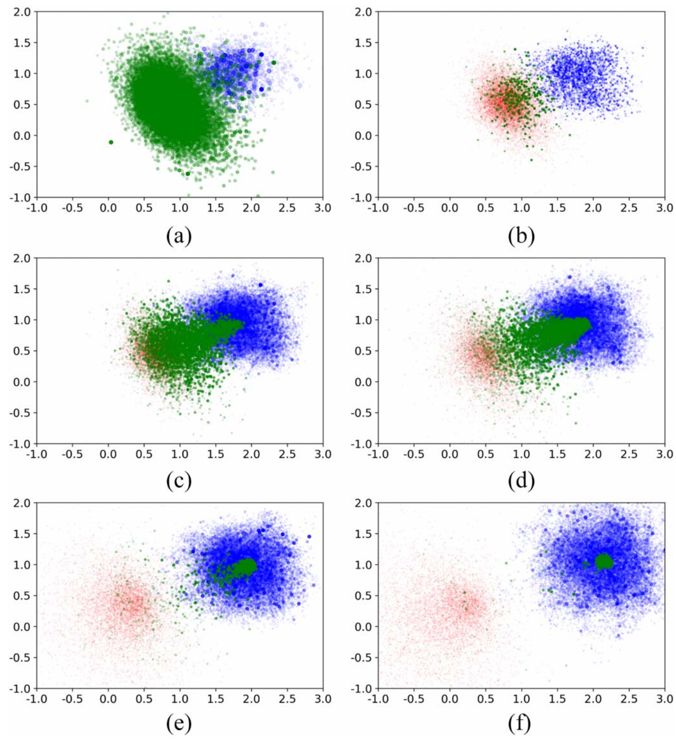


Fig. 6. Evolution of users and news in the latent representation space. The news is colored with blue, users with `LIKE` records are colored with green, and users without `LIKE` records are colored with red. Representation distribution: (a) @original; (b) @init; (c) @round-10; (d) @round-20; (e) @round-50; and (f) @round-100.

any environmental hyperparameter present a two-stage or three-stage pattern.

Finding 1: Recommender system-driven online news communities naturally exhibit the "start-up"–"growth"–"maturity & decline" life-cycle under a variety of user group settings.

2) Latent Space Visualization: Fig. 6 visualizes the evolution of the latent representations of users and news in the default simulation setting using PCA [48]. The news is colored with blue, users with `LIKE` records are colored with green, and users without `LIKE` records are colored with red. The node size represents the number of `LIKE`s. Although the quantitative metrics exhibit a multistage pattern, the evolutionary trend of the latent space representations is consistent, i.e., users gradually differentiate into in-the-loop and out-the-loop. Users in the loop form a stable community with convergent interests, while users out of the loop present fragmented interests. During the interaction between the tenth and twentieth rounds, the users have basically completed the differentiation, which indicates that the growth phase is critical to user engagement.

Finding 2: Recommender system-driven online news communities would inevitably yield a convergence of community topics and lead to user differentiation. The key period that determines user engagement is the growth phase.

3) Textual Latent Space Explanation: Since we construct the latent space through PLMs, every vector in the space could be interpreted textually using words with similar embeddings. This helps to understand the evolution of the individual users through case studies, as a complement to the latent space visualization. We randomly select five users separately from in and out the recommendation feedback loop, and Table I presents the detailed textual explanation of the evolution of the ten users.[6] For users in the loop, the most similar words evolve toward more abstract, broader, and generalized nouns, e.g., from "Actors" to "Job" and from "Oslo" to "Norway" and "Europe." The evolutionary speed of different users varies, but all converge by the fiftieth round. This phenomenon reflects the migration of user preference from the personalized niche topics to the trending topics that are widely discussed on the platform, as a consequence of continuously interacting with the recommender system. For users out of the loop, their representations shift slightly, but always focus on specific and personalized topics. For instance, users #6 and #10 keep their interests toward "Athletes," "Tea," "Bill," and "Insect," respectively, throughout the simulation.

Beyond individual case study, constructing an explainable latent space with PLMs also aids in understanding the evolution of group attention. By encoding topics into latent vectors, we can compare the users' preferences for various topics. Table II presents the distribution of initial user interests across seven topics (users with cosine similarity less than 0.8 for all topics are categorized as "else"), and the assimilation rate of initial members of each topic at the end of the simulation (assimilation refers to the evolution of user interests toward the common concern topics). In the initial data distribution, "local news" emerged as the most popular news topic, with others such as "politics" and "science" being relatively less attractive. The assimilation ratio among different topics varies, with a difference of over 10% between the highest and lowest; most topics have an assimilation ratio around 43%. "Local news" has the highest assimilation ratio, reflecting the impact of user scale on public topics.

Further, we visualize the transformation in topic attention among users before and after the simulation, as shown in Fig. 7. Users from all topics exhibit both inflows and outflows, with the overall trend being a convergence toward "local news." However, "local news" also experiences the highest user outflow, suggesting that alignment of user interests with public topics does not necessarily guarantee a satisfactory user experience.

Finding 3: In recommender system-driven online news communities, users' personalized interests are assimilated in the process of continuous interaction with the recommendation system.

C. Life-Cycle Analysis

To the best of our knowledge, SimuLine is the first work to validate the lifecycle of NREs through simulation. In this section, we discuss the characteristics of each phase in detail.

1) Start-Up Phase: Corresponding to the first ten rounds, this phase addresses the cold-start user issue, i.e., involving the

[6] For each user, we retrieve the top three nouns with the highest cosine similarity from the wordbase which contains 200 000 Norwegian subwords and translate them into English as the textual explanation of the target representation.

TABLE I
TEXTUAL EXPLANATION OF THE LATENT REPRESENTATION EVOLUTION OF SIX SYNTHETIC USERS, WHERE THE FIRST THREE USERS PARTICIPATED IN THE RECOMMENDATION FEEDBACK LOOPS, AND THE LAST THREE USERS FAILED TO GET INVOLVED. IN-THE-LOOP USERS GRADUALLY DEVELOP A CONVERGENT INTEREST IN BIG TOPICS, WHILE OUT-THE-LOOP USERS RETAIN THEIR PERSONALIZED INTERESTS IN MINORITY TOPICS

User		Initialization	Fifth Round	Tenth Round	Twentieth Round	Fiftieth Round	Hundredth Round
In the loop	User #1	November February Actors	Autumn Manager Child	Year Example Autumn	Job Year Europe	Job Year Europe	Job Year Europe
	User #2	Oslo School Post Office	Oslo School Post Office	Mountain Norway Stations	Mountains Wood Norway	Europe Time Reason	Europe Time Reason
	User #3	Nuts Christian Cello	Nuts Christian Cello	Sky Knowledge Musician	Time Reason Example	Year Reason Example	Year Reason Example
	User #4	Archive Ceramics Agriculture	Agriculture Paper Wine	Fuel Keramikk Vineyards	Norway Decade History	Oslo Year Norwegian	Europe Year History
	User #5	Motorsport Inventory Rocket	Association Motorsport Qualification	Racetrack Soyuz Camera	Formula Tournament Vehicle	Norway Year Event	Norway Year Example
Out the loop	User #6	Hometown Athletes Finals	Hometown Athletes Prophets	Athletes Hometown Ferret	Athletes Hometown Ferret	Athletes Ferret Ropes	Athletes Ferret Knife
	User #7	Runners Champions Tea	Runners Champions Tea	Runners Tea Karate	Tea Runner Karate	Tea Runner Vascular	Tea Vascular Frozen
	User #8	Money Countryman Bill	Countryman Money Bill	Billing Countryman Money	Billing Countryman Police	Billing Countryman Police	Billing Countryman Police
	User #9	Athletes Winter Games Butterfly	Relay Butterfly Ceramics	Athletics Relay Agriculture	Winter Games Relay Corn	Relay Corn Biodiversity	Camouflage Rematch Aircraft
	User #10	Insect Doping Crash	Insect Test Accident	Insect Test Silk	Insect Doping Polo	Insect Doping Polo	Insect Tennis Crash

TABLE II
TOPIC DISTRIBUTION OF USERS' INITIAL INTERESTS AND ASSOCIATION RATIO OF EACH TOPIC

Topic	Num User	Assimilation Ratio (%)
Finance	647	37.40
Politics	343	38.48
Sports	1696	42.69
Entertainment	1687	43.39
Science	382	42.93
Life	840	42.26
Local	3065	47.77
Else	1340	42.69

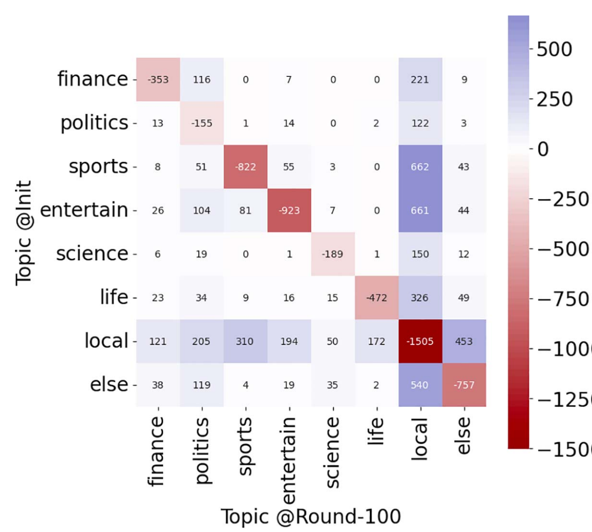


Fig. 7. Transformation of user interests among topics. The vertical axis represents the user's initial interest, and the horizontal axis represents the user's interest after 100 rounds of simulation.

users in the recommendation feedback loop via random recommendations. The LIKEs in this stage are mainly driven by the news quality, leading to the positive correlation between quality and popularity as shown in Fig. 5(c). The development of this phase is catalyzed by two main factors: 1) the quality feedback loop, which refers to the mutual promotion between quality and popularity based on the positive correlation; and 2) the interest-quality confusion, which refers to that until enough data are accumulated to accurately estimate user interests, the recommendation algorithm confuses the quality-driven LIKEs as interest-driven. This helps the popular content creators to obtain excessive exposure and further enhances the quality feedback loop, resulting in the decline in similarity between the latent vectors of users and their liked news as shown in Fig. 5(d). Most users can benefit from the improved news quality, decreasing the Gini index of user LIKEs as shown in Fig. 3(b).

Finding 4: In start-up phase, NREs accumulate data for estimating user interests from random recommendations and high-quality news, which can address the cold-start user issue. The quality feedback loop and interest-quality confusion contribute to the emergence of highly popular content creators via excessive exposure.

2) Growth Phase: With the accumulation of data, the recommendation model becomes increasingly accurate in estimating user interests, and the `LIKE` behaviors are more interest-driven than quality-driven. The correlation between quality and popularity gradually weakens, resulting in the end of the quality feedback loop around the tenth round. As shown in Fig. 6(c), the density of news around each in-the-loop user is uneven, with a higher density in the direction toward the mainstream news topic and smaller densities in other directions. As a result, the liked news is statistically closer to the mainstream news topic compared with the users in the latent space, and the user interests are gradually cultivated toward the mainstream news topic through continuous interactions with the recommender system. On the contrary, the out-the-loop users are trapped in the deadlock of "no `LIKE`—not covered by the algorithm—poor recommendation quality—no `LIKE`." They still give `LIKE`s occasionally, but the time-bound data are too sparse for the recommendation models to accurately estimate their interests and stimulate more `LIKE`s in future rounds. More frequent and balanced `LIKE`s stimulate the growth of news quality, but due to the declining popularity of high-quality news, the quality weighted by the number of `LIKE`s remains stable as shown in Fig. 4(b)–4(d). As the quality feedback loop terminates, content creators cannot receive excessive attention and thus the news quality falls back, resulting in that quality-sensitive users stop giving `LIKE`s which presents as the drop in user coverage as shown in Fig. 5(a).

Finding 5: In the growth phase, in-the-loop users evolve toward common topics under the effect of distribution bias, while out-the-loop users are trapped in a deadlock, resulting in user differentiation. The growing accurate algorithmic recommendation leads to the ending of the quality feedback loop, resulting in the reduced coverage of quality-sensitive users.

3) Maturity & Decline Phase: At around the twentieth round, the NRE enters the maturity & decline phase as most of the key metrics are relatively stable. In this phase, the in-the-loop users are dynamically kept in the bubble of common topics. Although their interests may shift toward the edge of the bubble due to clicking on some diverse news, they would be soon back to the center due to the news articles around them are highly similar to the center of the bubble. As shown in Fig. 4(a) and 4(b), the Gini index of `LIKE`s on news is high but that on content creators is low, indicating that the news even created by the same creator presents highly varying popularity. In addition to the greedy creating mechanism, the process of news creation is highly stochastic and presents a natural tendency of expansion, which is reflected in Fig. 6 by the expansion of blue nodes and the Fig. 5(d) by a small decrease in user-news similarity. The declining user-news similarity further leads to the quitting of interest-sensitive users as shown in Fig. 5(a).

Finding 6: In the maturity & decline phase, in-the-loop users share common topics, around which content creators publish diverse news. With the constraints of the recommender system, the NRE maintains the stability of the community but along with slow loss of interest-sensitive users.

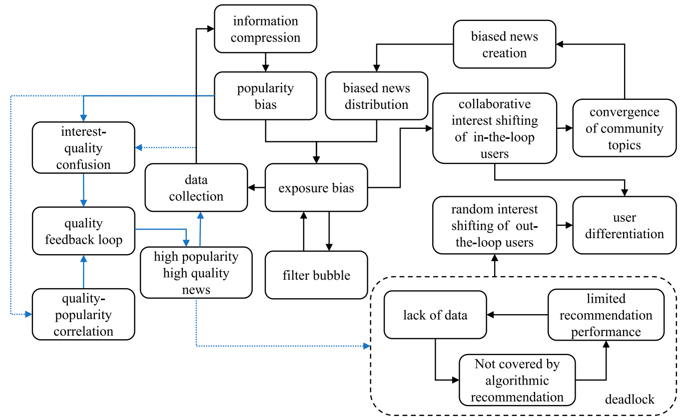


Fig. 8. Relationship graph of news recommendation ecosystem evolution. The blue arrows occur in the start-up phase. The dotted arrows represent negative effects.

D. Key Factors and Affecting Mechanisms

Fig. 8 illustrates the key factors and affecting mechanisms of NRE evolution, from which we could find that the re-emerging exposure bias and the deadlock are the direct causes of the different evolution trends between in-the-loop users and out-the-loop users, resulting in the user differentiation and convergence of topics.

The re-emerging exposure bias is caused by multiple factors.

1) Popularity Bias: According to the information theory [53], training recommendation algorithms can be interpreted as a process of information compression, inevitably leading to the popularity bias, in which high-frequency items are encoded and recommended more efficiently. This phenomenon impacts the behavior of in-the-loop users on two levels. First, news articles with higher popularity have a higher probability of appearing in a user's recommendation list. Second, even within the same recommendation list, popular news articles, due to more extensive training, align more closely with user interests and are thus more likely to be read. Consequently, both the recommendations and reading behaviors of in-the-loop users exhibit exposure bias.

2) Biased News Distribution: Due to the profit-seeking nature of content creators, they are more motivated to create news around topics of high public interest, which leads to biased news distribution with a decreasing density from common topics to personalized topics in the latent space. This means that even with random news article recommendations, common topics present a higher exposure probability than personalized topics, creating topic-level exposure bias.

3) Filter Bubble: Algorithmic recommendations, by only providing news that aligns with user interests, restrict users' exposure to a limited range of topics, thus creating a filter bubble. The density distribution of news in the latent space is continuous; therefore, even if the global distribution is biased, this bias is hard to perceive within a localized range. This makes the exposure bias caused by popularity bias and biased news distribution covert from the user's viewpoint. Over time, under the influence of exposure biases in recommendation and

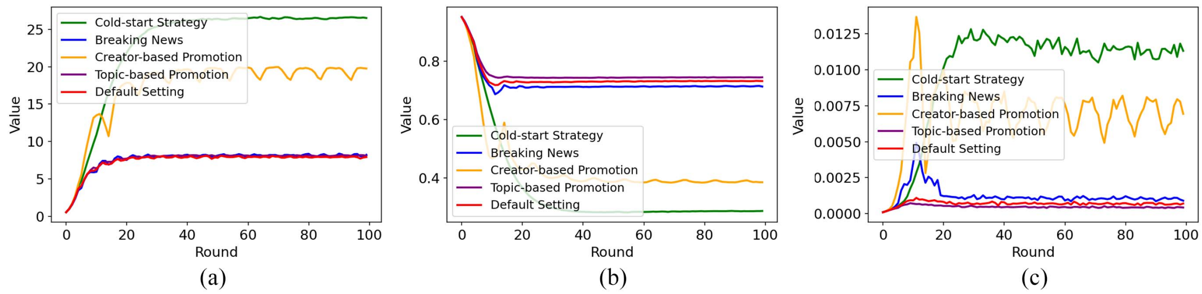


Fig. 9. User-related metrics under different recommender system designs, including (a) averaged number of LIKEs, (b) Gini index of users, and (c) Jaccard index of users.

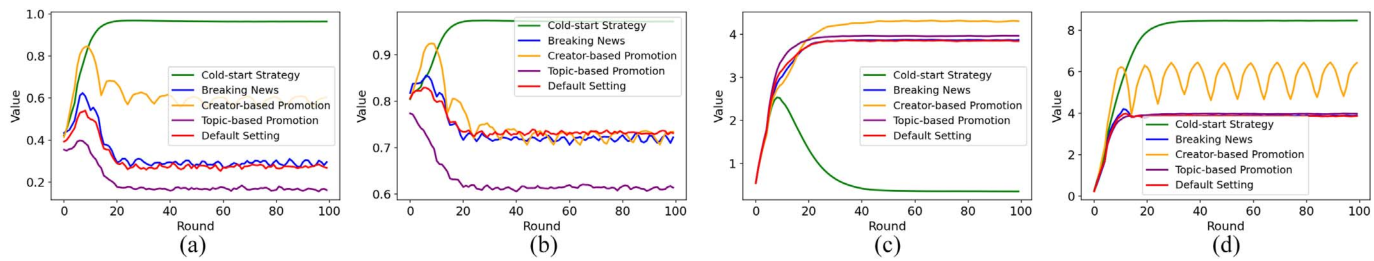


Fig. 10. Content creator-related metrics under different recommender system designs, including Gini index of (a) content creators, (b) news, (c) average quality of active news, and (d) LIKE-weighted quality of active news.

reading behaviors, the interests of in-the-loop users gradually shift toward common interest topics. In contrast, out-the-loop users are less affected. This difference in interest evolution leads to user differentiation. As in-the-loop users, the main group producing interactive behaviors present assimilating interests, and the creators also shift their focus to the converging common interest topics.

Besides, the popularity bias presents different effects on the article quality of recommended news in different evolutionary phases: 1) In the start-up phase, due to the presence of interest-quality confusion and quality-popularity correlation, popularity bias tends to favor high-quality news, essentially acting as a quality bias. This stage promotes the cultivation of high-popularity and high-quality news articles, which are widely read by various audiences and drive initial user interactions based on quality. This is crucial in overcoming the deadlock challenge faced by out-the-loop users. 2) In the growth phase, the accumulation of user interaction data progressively weakens the interest-quality confusion. This refinement allows algorithms to more accurately estimate user interests, consequently beginning to disrupt the quality-popularity correlation. During this phase, while the popularity bias still exhibits a quality bias effect, this effect is gradually diminishing. 3) In the maturity & decline phase, with the ecosystem stabilizing, the quality-popularity correlation completely dissipates. In this stage, the popularity bias no longer yields any extra quality gains, pivoting toward purely promoting high-popularity news. This evolution illustrates the dynamic nature of algorithmic recommendations and their changing influence over time.

Finding 7: Popularity bias, biased news distribution, and filter bubble together lead to exposure bias—the key factor affecting user differentiation and topic convergence. The high-popularity and high-quality news is crucial in breaking the deadlock of out-the-loop users.

E. Improving the Evolution via Recommender System Designing Strategies

This section explores how to utilize recommender system designing strategies to guide the evolution of NREs and thus achieve improved long-term utility from the perspective of metrics analysis and network analysis.

1) Cold Start: Illustrated by the green lines in Figs. 9, 10, and 11, we push the cold-start news to users who used to like the corresponding content creators instead of recommending randomly. This method attempts to form a stable cross-round exposure relationship between users and content creators, which enhances the quality feedback loop in the start-up phase. However, this approach leads to a serious monopoly. The content creators who have not achieved a first-mover advantage are suppressed by the quality feedback loop, damaging the algorithmic coverage and average quality of news, which in turn makes the diversity of the ecosystem severely challenged.

2) Breaking News: Illustrated by the blue lines in Figs. 9, 10, and 11, we add the 20 most liked but unrecommended news in the previous round to the recommendation list, and correspondingly reduce the number of algorithmically recommended news by 20. Relying on the positive correlation between popularity and quality, this method could provide users with news of higher quality, while ensuring user exploration and avoiding the monopoly due to the absence of a stable exposure relationship between users and content creators. From the perspective of

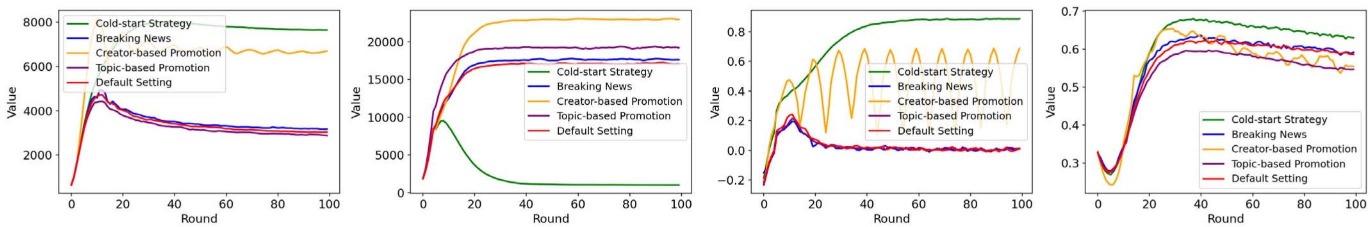


Fig. 11. Recommender system-related metrics under different recommender system designs, including the number of (a) users, (b) news covered by the algorithmic recommendation, (c) correlation coefficient between popularity and quality, and (d) cosine similarity between users and their liked news.

TABLE III
NETWORK ANALYSIS OF USER COLLABORATIVE RELATIONSHIPS UNDER THE INFLUENCE OF DIFFERENT RECOMMENDATION SYSTEMS

	Default	Cold start	Breaking News	Creator Promotion	Topic Promotion
Num. Augment Relations	469 942	17 134 192	5 068 400	3 598 371	176 680
Network Diameter	12	20	8	24	13
Avg. Shortest Path Length	3.13	2.45	1.97	4.48	3.61
Avg. Two-hop Reachable	325.11	7081.87	2497.85	1676.16	97.38
Num. of Community	21	3	11	11	66
Num. of Islands	4301	897	2557	1457	4759
Avg. Top-3 Community Size	836.67	3034.33	1287.33	1485.67	491.33
Modularity	0.1181	0.0869	0.1088	0.0528	0.1750
In-community User Similarity	0.9247	0.8178	0.1392	0.4778	0.7653

exploitation and exploration, reading breaking news could also be regarded as a kind of user exploration, which can help mitigate the negative effects of the filter bubble. However, this method could not prevent the collapsing of the critical correlation between popularity and quality as discussed in Section IV-D, resulting in the declining effectiveness of recommending breaking news.

3) Promotion: Illustrated by the yellow and purple lines in Figs. 9, 10, and 11, we split 20 quotas of news from algorithmic recommendations to the promotion of randomly selected content creators or topics. The promoted content creators or topics are reset per ten rounds. The content creator-based promotion builds a stable exposure relationship and leverages the quality feedback loop to cultivate high-popularity and high-quality news. But different from the cold-start strategies, the promotion could be terminated proactively before the current quality feedback loop cultivates a harmful monopoly, thereby enhancing the user experience and creator creativity. As an independent news-feeding channel, it could mitigate the negative effects of the filter bubble. Besides, by rebuilding the quality feedback loop, it also directs the effect of popularity bias toward beneficial recommendations for high-quality news. As we select the topics randomly when performing the topic-based promotion, the trending topics have an equal chance to be promoted as the personalized topics, so that the promotion has a relatively larger impact on the personalized topics considering their relatively lower exposures. Theoretically, it can be used to improve the engagements of out-the-loop users, but due to the quality of promoted news cannot always be promised, it is difficult to convert the exposures into LIKEs, resulting in limited effectiveness.

Finding 8: Among the typical recommender system designing strategies, periodic content creator-based promotion is the most effective in keeping platform active. By actively building the quality feedback loop, it could create waves of high-popularity and high-quality news topics throughout the NREs, and meanwhile keep the monopoly under control via periodic reset.

Further, we employ network analysis to compare the impact of different recommendation strategies on user behaviors. Initially, we construct a user collaboration network based on the consistency of user LIKEs: If over 10% of the articles liked by user u are also liked by user v and vice versa, an edge between u and v is established.[7] This edge signifies a high degree of behavioral homogeneity between users u and v.

We analyze the constructed user collaboration network from two perspectives: network structure and community detection [54] in Table III. In terms of network structure, the number of augment relations represents the overall degree of user homogenization. The cold-start strategy leads to a remarkable level of homogenization due to the emergence of monopolies. Both the breaking news and creator promotion strategies significantly increase user homogenization as well. Only the topic promotion strategies notably decrease user homogenization, aligning with our previous analysis that topic promotion is inclined of recommending personalized topics. The network diameter and average shortest path length measure the network's centralization. Breaking news strategy, by recommending the same popular news to everyone, results in the strongest centralization. In contrast, creator promotion strategy, by building local relationships around the promoted creators, achieves a distinct decentralization effect.

[7]The choice of the threshold does not affect the final conclusion; we select a value that provide clear comparative results.

TABLE IV
RANKING ACCURACY OF DIFFERENT USER MODELING APPROACHES

	AUC (%)	MRR (%)	nDCG5 (%)	nDCG10 (%)
Multihot	31.89	28.01	30.80	34.55
Fully trained	67.47	61.62	61.24	66.95
Ours	66.42	61.06	60.80	65.88

TABLE V
EFFECTIVENESS VALIDATION OF IPS IN REDUCING USER MODEL BIAS

	Avg. News Popularity
Ground-Truth in Testing Dataset	4610.99
Top-1 Ranked w/o IPS	4675.62 (+1.40%)
Top-1 Ranked w/z IPS	4618.13 (+0.15%)

TABLE VI
VALIDATION OF THE IMPACT OF IPS ON USER MODEL ACCURACY

	AUC	MRR	nDCG5	nDCG10	Discount
w/o IPS	66.87	61.34	60.91	66.03	-
w/z IPS	66.42	61.06	60.80	65.88	−0.38%

In terms of community detection, cold-start, breaking news, and creator promotion lead to varying degrees of community merging, consistent with the homogenization analysis results. The topic promotion strategy cultivates more personalized-interest communities, enhancing overall diversity. Merged communities exhibit lower modularity, indicating that user behaviors within them are not completely uniform; whereas personalized communities show higher uniformity. After introducing these strategies, user similarity within communities decreases, correlating with the personalized degree of applied strategy. This also highlights the superiority of deep learning algorithms in personalized recommendation.

Finding 9: Cold-start, breaking news, and creator promotion strategies lead to the homogenization of user behaviors; topic promotion can be employed to strengthen personalization and diversification; creator promotion presents to be an efficient tool in decentralizing.

F. System Validation

In this section, we validate the performance of the proposed user modeling method and compare the findings conducted from our simulating experiments with those of existing works, to illustrate the validity and advancement of SimuLine.

1) Validation of the User Modeling: To validate the accuracy of user modeling in SimuLine, we compare its ranking accuracy with two widely used approaches in the existing literature and discuss their applicability in news recommendation simulations. The first approach involves the manual definition of latent space, leveraging expert knowledge or preexisting labels [22], [23]. The second approach employs deep learning models to derive user and news embeddings as latent vectors from historical data [11], [40]. For both methods, we implement a representative baseline as follows.

a) Multihot: We utilize one-hot encoding for all categories and keywords in the dataset. News articles are represented by multihot encoded labels, and the average of the latent vectors of historically interacted news is used as the users' latent vectors.

b) Fully trained: The user and news embeddings are trained with the BPR [49] algorithm. Evaluation employs standard ranking metrics such as AUC, MRR, and nDCG, with results presented in Table IV.

As illustrated in Table IV, fully trained, which leverages collaborative information between users and news, achieves the highest ranking accuracy. However, its latent space, grounded in collaborative data, is not suitable for sampling news articles. This space represents specific historical interaction patterns, but predicting the dissemination of a new article is not feasible at its creation. Consequently, attempting to sample news within this framework leads to a causal contradiction, making this approach impractical for news recommendation simulation purposes. By introducing PLMs, SimuLine resolves the aforementioned logical dilemma and achieves comparable performance. Developed on extensive corpora, PLMs model the context and semantics of textual content, aligning closely with the news creation and reading behaviors. This alignment makes PLMs more suitable for news recommendation simulation scenarios. Compared to the previous two methods, the multihot approach falls short in accurately modeling user behavior patterns, thereby reducing its reliability for supporting precise simulations.

Further validation of IPS demonstrates its effectiveness in reducing biases in user modeling. Influenced by the online recommender system, the dataset presents inherent exposure bias, which means unequal occurring frequencies for users and news. This often results in popularity bias in user models. Table V illustrates this by comparing the popularity (measured by the frequency of occurrence in the dataset) of real user-interacted news in the test set with the top-ranked news by the user model. Without IPS, the model shows a 1.4% popularity bias. With IPS, this bias significantly drops to 0.15%, a substantial tenfold reduction. Table VI compares the ranking accuracy of the user model with and without IPS. The introduction of IPS leads to a slight decrease in accuracy by just 0.38%, indicating that IPS effectively reduces bias with negligible impact on accuracy.

2) Validation of the Findings: The simulation experiments driven by SimuLine yield many intriguing findings. These findings show consistency with the existing literature, which underscores the reliability of SimuLine.

Findings 1–3 verify the existence of a multiphase life-cycle in the evolution of online news communities and highlight the phenomena of topic convergence, user differentiation, and interest assimilation during the evolution of communities, which is consistent with the findings of extensive empirical studies of online communities [7], [55], [56].

Findings 4–7 attempt to explain the evolutionary process of online news communities, which also validate and extend the conclusions of existing works. Regarding the relationship between popularity and quality, Zhao et al. [16] and Ciampaglia et al. [17] pointed out that under certain conditions, high quality leads to high popularity. In our Findings 4 and 5, we discuss the quality feedback loops in the start-up and growth phases, further revealing how this correlation establishes and expires. Liu et al. [5] and Bountouridis et al. [21] discussed the impact of recommendation algorithms on view convergence and user fragmentation, which are verified in our Finding 5. Yao et al. [22] found that the popularity bias of recommendation results is mitigated as more interactions occur, which is consistent with our Finding 6. Chaney et al. [18] proposed that recommender systems amplify the homogeneity of user content consumption because of algorithmic confounding, which is similar to our arguments about exposure bias in Finding 7.

Findings 8 and 9 explore how recommender system designs could contribute to the prosperity of online communities. Ciampaglia et al. [17] and Jiang et al. [13] highlighted the impact of user exploration on improving long-term utilities. We further refine platform-guided user exploration into strategies such as cold-start, breaking news, and promotions, which are tested and discussed in detail. The results show that a well-designed user exploration strategy can lead to stable improvement, but inappropriate explore strategies may play negative roles.

V. Conclusion and Future Work

This article presents SimuLine, a simulation platform to dissect the evolution of news recommendation ecosystems, and provides a detailed analysis of the evolutionary process. SimuLine constructs a latent space well reflecting the human behaviors, based on which we simulate the NREs via agent-based modeling. SimuLine dissects the lifecycle of the NRE evolution, which consists of the start-up, growth, and maturity & decline phases. We analyze the characteristics of each phase and propose a relationship graph illustrating the key factors and affecting mechanisms. In the end, we explore the impacts of recommender system designing strategies, including the utilization of cold-start news, breaking news, and promotion, on the evolutionary process.

In the future, we will consider supporting the textual content generation of the synthetic news and the action modeling of social network activities for more powerful simulation. Besides, SimuLine also benefits the evaluation of recommendation algorithms. In particular, some debiasing recommendation algorithms [46] are recently proposed aiming at handling the exposure bias, which is the direct cause of user differentiation and topic convergence. Since this article focuses on the systematic design of recommender systems rather than on specific recommendation algorithms, we leave this issue as an open topic and hope that SimuLine could facilitate future research in this direction.

References

[1] M. Li and L. Wang, "A survey on personalized news recommendation technology," *IEEE Access*, vol. 7, pp. 145861–145879, 2019.
[2] C. Wu, F. Wu, Y. Huang, and X. Xie, "Personalized news recommendation: Methods and challenges," *ACM Trans. Inf. Syst. (TOIS)*, vol. 41, no. 1, pp. 1–50, 2022.
[3] G. Merchant et al., "Click 'like' to change your behavior: A mixed methods study of college students' exposure to and engagement with Facebook content designed for weight loss," *J. Med. Internet Res.*, vol. 16, no. 6, 2014, Art. no. e3267.
[4] C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum, "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing," *Nature Hum. Behav.*, vol. 1, no. 4, pp. 1–10, 2017.
[5] P. Liu, K. Shivaram, A. Culotta, M. A. Shapiro, and M. Bilgic, "The interaction between political typology and filter bubbles in news recommendation algorithms," in *Proc. Web Conf.*, 2021, pp. 3791–3801.
[6] D. C. Mueller, "A life cycle theory of the firm," *J. Ind. Econ.*, vol. 20, no. 3, pp. 199–219, Jul. 1972.
[7] A. Iriberri and G. Leroy, "A life-cycle perspective on online community success," *ACM Comput. Surveys (CSUR)*, vol. 41, no. 2, pp. 1–29, 2009.
[8] H. Fu and S. Oh, "Online community development in the early stages: The life cycle model application to medical sciences stack exchange," *Aslib J. Inf. Manage.*, vol. 74, no. 6, pp. 1214–1232, 2022.
[9] A. S. Sha, A. Haller, and Y. Shi, "Effects of label usage on question lifecycle in q&a community," *Proc. 30th European Conf. Inf. Syst.*, Timisoara, Romania, Jun. 18–24, 2022.
[10] E. Stavinova, A. Grigorievskiy, A. Volodkevich, P. Chunaev, K. Bochenina, and D. Bugaychenko, "Synthetic data-based simulators for recommender systems: A survey," 2022, *arXiv:2206.11338*.
[11] X. Luo, Z. Liu, S. Xiao, X. Xie, and D. Li, "MINDSim: User simulator for news recommenders," in *Proc. ACM Web Conf.*, 2022, pp. 2067–2077.
[12] D. Geschke, J. Lorenz, and P. Holtz, "The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers," *Brit. J. Social Psychol.*, vol. 58, no. 1, pp. 129–149, 2019.
[13] R. Jiang, S. Chiappa, T. Lattimore, A. György, and P. Kohli, "Degenerate feedback loops in recommender systems," in *Proc. AAAI/ACM Conf. AI, Ethics, Soc.*, 2019, pp. 383–390.
[14] G. Aridor, D. Goncalves, and S. Sikdar, "Deconstructing the filter bubble: User decision-making and recommender systems," in *Proc. 14th ACM Conf. Recommender Syst.*, 2020, pp. 82–91.
[15] W. Sun, S. Khenissi, O. Nasraoui, and P. Shafto, "Debiasing the human-recommender system feedback loop in collaborative filtering," in *Companion Proc. World Wide Web Conf.*, 2019, pp. 645–651.
[16] Z. Zhao et al., "Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation," 2021, *arXiv:2109.07946*.
[17] G. L. Ciampaglia, A. Nematzadeh, F. Menczer, and A. Flammini, "How algorithmic popularity bias hinders or promotes quality," *Sci. Rep.*, vol. 8, no. 1, pp. 1–7, 2018.
[18] A. J. Chaney, B. M. Stewart, and B. E. Engelhardt, "How algorithmic confounding in recommendation systems increases homogeneity and decreases utility," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 224–232.
[19] K. Krauth et al., "Do offline metrics predict online performance in recommender systems?" 2020, *arXiv:2011.07931*.
[20] E. Lucherini, M. Sun, A. Winecoff, and A. Narayanan, "T-RECS: A simulation tool to study the societal impact of recommender systems," 2021, *arXiv:2107.08959*.
[21] D. Bountouridis, J. Harambam, M. Makhortykh, M. Marrero, N. Tintarev, and C. Hauff, "Siren: A simulation framework for understanding the effects of recommender systems in online news environments," in *Proc. Conf. Fairness, Accountability, Transparency*, 2019, pp. 150–159.
[22] S. Yao et al., "Measuring recommender system effects with simulated users," 2021, *arXiv:2101.04526*.
[23] M. Zhou, J. Zhang, and G. Adomavicius, "Longitudinal impact of preference biases on recommender systems' performance," in *Kelley School Bus. Res. Paper*, no. 2021-10, 2021.
[24] X. Ma et al., "Entire space multi-task model: An effective approach for estimating post-click conversion rate," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2018, pp. 1137–1140.
[25] Y. Saito, "Unbiased pairwise learning from biased implicit feedback," in *Proc. ACM SIGIR Int. Conf. Theory Inf. Retrieval*, 2020, pp. 5–12.
[26] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.
[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
[28] C. Allen and T. Hospedales, "Analogies explained: Towards understanding word embeddings," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 223–231.
[29] K. Ethayarajh, D. Duvenaud, and G. Hirst, "Towards understanding linear word analogies," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3253–3262.
[30] M. del Carmen Rodríguez-Hernández, S. Ilarri, R. Hermoso, and R. Trillo-Lado, "DataGenCARS: A generator of synthetic data for the evaluation of context-aware recommendation systems," *Pervasive Mob. Comput.*, vol. 38, pp. 516–541, 2017.

[31] M. Jakomin, T. Curk, and Z. Bosnić, "Generating inter-dependent data streams for recommender systems," *Simul. Modell. Pract. Theory*, vol. 88, pp. 1–16, 2018.

[32] M. Slokom, "Comparing recommender systems using synthetic data," in *Proc. 12th ACM Conf. Recommender Syst.*, 2018, pp. 548–552.

[33] D. Rohde, S. Bonner, T. Dunlop, F. Vasile, and A. Karatzoglou, "RecoGym: A reinforcement learning environment for the problem of product recommendation in online advertising," 2018, *arXiv:1808.00720*.

[34] X. Chen, S. Li, H. Li, S. Jiang, Y. Qi, and L. Song, "Generative adversarial user model for reinforcement learning based recommendation system," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1052–1061.

[35] E. Ie et al., "RecSim: A configurable simulation platform for recommender systems," 2019, *arXiv:1909.04847*.

[36] M. Mladenov et al., "RecSim NG: Toward principled uncertainty modeling for recommender ecosystems," 2021, *arXiv:2103.08057*.

[37] J.-C. Shi, Y. Yu, Q. Da, S.-Y. Chen, and A.-X. Zeng, "Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 4902–4909.

[38] J. Huang, H. Oosterhuis, M. De Rijke, and H. Van Hoof, "Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems," in *Proc. 14th ACM Conf. Recommender Syst.*, 2020, pp. 190–199.

[39] J. McInerney, E. Elahi, J. Basilico, Y. Raimond, and T. Jebara, "Accordion: A trainable simulator for long-term interactive systems," in *Proc. 15th ACM Conf. Recommender Syst.*, 2021, pp. 102–113.

[40] X. Zhao, L. Xia, L. Zou, H. Liu, D. Yin, and J. Tang, "UserSim: User simulation via supervised generative adversarial network," in *Proc. Web Conf.*, 2021, pp. 3582–3589.

[41] W. Wang, "Learning to recommend from sparse data via generative user feedback," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 5, 2021, pp. 4436–4444.

[42] J. S. Park, J. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein, "Generative agents: Interactive simulacra of human behavior," in *Proc. 36th Annu. ACM Symp. User Interface Softw. Technol.*, 2023, pp. 1–22.

[43] L. Wang et al., "RecAgent: A novel simulation paradigm for recommender systems," 2023, *arXiv:2306.02552*.

[44] A. Zhang et al., "On generative agents in recommendation," 2023, *arXiv:2310.10108*.

[45] B. Heinzerling and M. Strube, "BPEmb: Tokenization-free pre-trained subword embeddings in 275 languages," in *Proc. 11th Int. Conf. Lang. Resour. Eval. (LREC)*, 2018.

[46] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," 2020, *arXiv:2010.03240*.

[47] G. W. Imbens and D. B. Rubin, *Causal Inference in Statistics, Social, and Biomedical Sciences*. Cambridge, U.K.: Cambridge Univ. Press, 2015.

[48] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, vol. 4, no. 4.

[49] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," 2012, *arXiv:1205.2618*.

[50] R. C. Wilson, A. Geana, J. M. White, E. A. Ludvig, and J. D. Cohen, "Humans use directed and random exploration to solve the explore–exploit dilemma," *J. Exp. Psychol.: General*, vol. 143, no. 6, p. 2074, 2014.

[51] S. J. Gershman, "Deconstructing the human algorithms for exploration," *Cognition*, vol. 173, pp. 34–42, 2018.

[52] J. A. Gulla, L. Zhang, P. Liu, Ö. Özgöbek, and X. Su, "The Adressa dataset for news recommendation," in *Proc. Int. Conf. Web Intell.*, 2017, pp. 1042–1048.

[53] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Techn. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[54] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Statist. Mech.: Theory Exp.*, vol. 2008, no. 10, 2008, Art. no. P10008.

[55] S. Mohseni, E. Ragan, and X. Hu, "Open issues in combating fake news: Interpretability as an opportunity," 2019, *arXiv:1904.03016*.

[56] O. Trach and S. Fedushko, "Determination of measures of counteraction to the social-oriented risks of virtual community life cycle organization," in *Proc. Conf. Comput. Sci. Inf. Technologies*, New York, NY, USA: Springer, 2019, pp. 680–695.

Guangping Zhang received the B.Eng. degree computer science from the School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, China, in 2020. He is working toward the Ph.D. degree software engineering with the School of Computer Science, Fudan University, Shanghai, China.

His research interests include computer-supported cooperative work (CSCW), social computing, recommender systems, and human–computer interaction (HCI).

Dongsheng Li (Senior Member, IEEE) received the B.E. degree from the University of Science and Technology of China, Hefei, Anhui Province, China in 2007, and the Ph.D. degree from Fudan University, Shanghai, China, in 2012.

He has been a Principal Research Manager with Microsoft Research Asia (MSRA), Shanghai, China, since 2020. Before joining MSRA, he has been a Research Staff Member with IBM Research, Shanghai, China, since 2015. He is also an Adjunct Professor with the School of Computer Science, Fudan University. His research interests include recommender systems and machine learning applications.

Dr. Li's work on cognitive recommendation engine won the 2018 IBM Corporate Award.

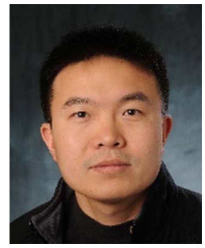
Hansu Gu (Member, IEEE) received the B.S. degree in computer science from Fudan University, Shanghai, China, in 2008, and the Ph.D. degree in electrical engineering from the University of Colorado Boulder, Boulder, CO, USA, in 2013.

He is currently a Senior Applied Scientist with Amazon.com, Seattle, WA, USA, working on Amazon search. His research interests include recommender systems and natural language processing (NLP).

Tun Lu (Member, IEEE) received the Ph.D. degree in computer science from Sichuan University, Sichuan Province, China, in 2006.

He was a Visiting Professor with the HCI Institute, Carnegie Mellon University, Pittsburgh, PA, USA, from 2014 to 2015. He is currently a Full Professor with the School of Computer Science, Fudan University, Shanghai, China. He has published more than 80 peer-reviewed publications in prestigious conferences and journals such as CSCW, CHI, UbiComp, NuerIPS, WWW, SIGIR, IEEE Transactions on Knowledge and Data Engineering, and IEEE Transactions on Multimedia. His research interests include computer-supported cooperative work (CSCW), social computing, recommender systems, and human–computer interaction (HCI).

Dr. Lu is a Distinguished Member of the China Computer Federation (CCF) and a member of ACM. He is the Secretary General of the CCF Technical Committee of Cooperative Computing and an Executive Member of the SIGCHI China Chapter. He received the Best Paper Award of CSCW'15 and the Honorable Mention Award of CSCW'18. He has been active in professional services by serving as the PC Co-Chair (e.g., CollaborateCom 2021 and 2022, BESC 2022, Chinese CSCW 2017–2022, and CSCWD 2010), the Associate Chair (e.g., CHI 2019, 2020, and 2022, and CSCW 2019 and 2020), an Associate Editor of *International Journal of Cooperative Information Systems*, an Editorial Board Member of *Human-Centric Intelligent Systems*, and a Guest Editor and a Reviewer for many well-known journals and conferences.



Li Shang (Member, IEEE) received the Ph.D. degree from Princeton University, Princeton, NJ, USA.

He was the Deputy Director and the Chief Architect of Intel Labs China, Beijing, China and an Associate Professor with CU Boulder, Boulder, CO, USA. He is a Professor with the School of Computer Science, Fudan University, Shanghai, China. His research interests include computer systems, human-centered computing, machine learning, and VLSI & EDA, with over 100 publications and multiple best paper awards and nominations.

Dr. Shang was a recipient of NSF Career Award.

Ning Gu (Member, IEEE) received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1995.

He is currently a Professor with the School of Computer Science, Fudan University, Shanghai, China. His research interests include human-centered cooperative computing, computer-supported cooperative work (CSCW) and social computing, and human–computer interaction.