REST: Debiased Social Recommendation via Reconstructing Exposure Strategies

RUICHU CAI, Guangdong University of Technology and Guangdong Provincial Key Laboratory of Public Finance and Taxation with Big Data Application, China
FENGZHU WU, ZIJIAN LI, JIE QIAO, and WEI CHEN, Guangdong University of Technology, China
YUEXING HAO, Cornell University, USA
HAO GU, Tencent Technology (SZ) Co., Ltd, China

The recommendation system, relying on historical observational data to model the complex relationships among users and items, has achieved great success in real-world applications. Selection bias is one of the most important issues of the existing observational data-based approaches, which is actually caused by multiple types of unobserved exposure strategies (e.g., promotions and holiday effects). Though various methods have been proposed to address this problem, they are mainly relying on the implicit debiasing techniques but not explicitly modeling the unobserved exposure strategies. By explicitly Reconstructing Exposure Strategies (REST), we formalize the recommendation problem as the counterfactual reasoning and propose the debiased social recommendation method. In REST, we assume that the exposure of an item is controlled by the latent exposure strategies, the user, and the item. Based on the above generation process, we first provide the theoretical guarantee of our method via identification analysis. Second, we employ a variational auto-encoder to reconstruct the latent exposure strategies, with the help of the social networks and the items. Third, we devise a counterfactual reasoning based recommendation algorithm by leveraging the recovered exposure strategies. Experiments on four real-world datasets, including three published datasets and one private WeChat Official Account dataset, demonstrate significant improvements over several state-of-the-art methods.

CCS Concepts: • Human-centered computing → Social recommendation; • Mathematics of computing → Causal networks;

Additional Key Words and Phrases: Recommendation system, social recommendation system, causal effect, variational auto-encoders

This research was supported in part by the National Key R&D Program of China (Grant No. 2021ZD0111501), National Science Fund for Excellent Young Scholars (Grant No. 62122022), Natural Science Foundation of China (Grants No. 61976052 and No. 62206064) and Open Foundation of Guangdong Provincial Key Laboratory of Public Finance and Taxation with Big Data Application (Grant No. 2019B121203012).

Authors’ addresses: R. Cai (Corresponding author), Guangdong University of Technology and Guangdong Provincial Key Laboratory of Public Finance and Taxation with Big Data Application, 100 Waihuanxi Road, 510006, Guangzhou, Guangdong, China; e-mail: cairuichu@gmail.com; F. Wu, Z. Li, J. Qiao, and W. Chen, Guangdong University of Technology, 100 Waihuanxi Road, 510006, Guangzhou, Guangdong, China; e-mails: fzwu97, leizigin, qiaojie.chn, chenweide-light@gmail.com; Y. Hao, Cornell University, 1423 MVR, Cornell, 116 Reservoir Avenue, 14853, Ithaca, New York, USA; e-mail: yh727@cornell.edu; H. Gu, Tencent Technology (SZ) Co., Ltd, Kejizhongyi Avenue Tencent Building, Hi-Tech Park, 518052, Shenzhen, Guangdong, China; e-mail: nickgu@tencent.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
1556-4681/2023/11-ART38 $15.00
https://doi.org/10.1145/3624986

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 2, Article 38. Publication date: November 2023.
1 INTRODUCTION

Recommendation system [9, 20, 23, 46, 54, 55] is an important technique in the world. It has been used for a wide range of applications, such as e-commerce [6, 29], search engines [2, 41], and e-resource services platforms [4]. Recently, the data-driven recommendation systems, which use historical data to model the complex relationships among users and items, have achieved a huge success and become mainstream [14, 15, 19, 30].

Selection bias is one of the key issues to the success of the data-driven recommendation systems [8, 40, 44]. Because the historical data are collected under multiple types of exposure strategies and are seriously biased. Give an example in the information flow application, on the one hand, the items recommended by the system will get a high chance to expose and further results in the bias of the collected historical data; on the other hand, users usually prefer to watch and rate the popular videos/news such that the recommended strategies will capture the trending videos/news and further increase the exposures. These biases of the historical data will lead to overestimating or underestimating the performance of the trained recommendation systems, harm the performance of the deployed recommendation systems, and even result in the well-known negative phenomenon named Information Cocoons [39].

To tackle the aforementioned selection bias problem, many researchers raise several debiased recommendation algorithms [31, 35, 45]. One of the mainstream approaches is the inverse propensity weighting (IPW)-based methods [21], e.g., the Empirical Risk Minimization framework [35] and the CausE method [3]. Recently, Feng et al. [43, 50, 56] employ the concept of causal effect to address the selection bias challenge. By viewing the exposed items (v) behind the dataset as the common cause to the exposure (e) and the rating level (r), we can rephrase the existing methods into the causal graph in Figure 1(b). This figure further explicitly models the user preference, i.e., the user preference will increase the exposure due to the recommended strategies and the rating level based on the user preference.

According to the aforementioned debiased methods, we can easily find that the core of them is to remove the effect of exposure and to estimate the outcome of rate level if item v is exposed to user u. However, in a large range of historical data, different exposure strategies could change dramatically, because the recommendation systems always try to capture the popular trending items or the changing user preference. Take Figure 1(a) as an example, a dataset is a collection of observations over a number of different promotions strategies, such as Baby’s Day, Valentine’s Day, Black Friday, and seasonal offer. Such several types of strategies would render the existing debiased methods fail as they assume a stationary exposure strategy. Furthermore, the strategies are usually independent of the user preference and the item properties, but ignoring the changing strategies will entangle the information of strategies with the users and items, which further leads to biased results. Please note that the stationary exposure strategy assumption can be equivalently viewed as the stationary propensity score over time. For example, applying the stationary exposure strategy assumption for reweighting-based methods equals providing the same sample weights for the flowers on Valentine’s Day and Baby’s Day, which is obviously unreasonable. To address this challenge, one straightforward solution is to take the exposure strategies into account. Hence, we can obtain the revised causal graph shown
Fig. 1. (a) The historical data are collected under different strategies, such as the promotion on Black Friday and the Valentine’s Day Gift Shop. Panels (b–d) denote different causal generative processes, where the gray nodes are the observed variables and the white nodes denote the latent variables. (b) The common causal graph of the existing methods implicitly leverages the stationary strategy assumption. Note that $u$ denotes user variables; $e$ denotes the exposure variables; $v$ denotes the item variables; $r$ denotes the rate level. Panels (c, d) are the causal generation process of the proposed method. Note that $s$ denotes the strategies variables and $G$ denotes the social networks. (c) The causal graph takes the latent strategy variables into account. (d) To address the counterfactual problem in the recommendation, we bring the social networks into the causal generation process, where the user variables are latent due to the complexity of the aggregation process of social networks.

Given the causal graph shown in Figure 1(c), the recommendation task can be seen as a counterfactual question that *What the rate level $r$ would be if an item $v$ is exposed to a user $u$ under strategy $s$?* This question is hard to answer, since we can only obtain the rate level from the exposed dataset. This difficulty can be solved if we can find a similar user who has given the rate level for the same item. But how to find such a similar user is another nontrivial task with the assumption that each user is independent unless the social networks (or local neighbors) are taken into consideration. So, we further propose another revised causal graph as shown in Figure 1(d). It is noted that we let the user variables be latent when taking social networks into account. This is because the interests of users are influenced by their neighbors, so the user variables become an aggregation of neighbors’ information and are too complex to be explicitly described.

Based on the causal graph shown in Figure 1(d), we provide a practical approach for debiased recommendation **Reconstructing Exposure Strategies (REST)** by modeling different strategies behind observed data. First, we assume that the data generative process of recommendation follows the causal graph shown in Figure 1(d). Second, We summarize the problem of the recommendation systems as the counterfactual reasoning problem and provide the identification analysis for theoretical guarantee. Third, based on this causal generative process, we devise a variational-based counterfactual reasoning method to successively reconstruct the user latent variables and the exposure strategy latent variables. Extensive experimental studies demonstrate that the proposed REST method outperforms the state-of-the-art recommendation methods (including the latest methods based on causal effect) on three published datasets and one real-world WeChat official accounts dataset.

The rest of the article is organized as follows. Section 2 reviews existing studies on recommendation systems, including social recommendation systems, causality-based recommendation systems, and recommendation systems using a generative model. In Section 3, we expound the causal generation process under latent strategies and social networks. We also elaborate on the details about how to model the aforementioned causal generation process and how to implement
the proposed model in Section 4. Section 5 presents the experimental results on four real-world datasets, including ablation analysis and visualization. Section 6 concludes the article.

2 RELATED WORKS

Our work is closely related to the recommendation systems in the causal view, the social recommendation systems, and the recommendation systems that are related to generative models. In this section, we review the works on these three kinds of recommendation systems.

To address the problem of selection bias, many researchers borrow the ideas of causal inference [21, 27]. Sharma et al. [37] estimate the causal effect of the recommendation system from observed data. Schnabel et al. [35] estimate the quality of a recommendation system with the help of the propensity-weighting method, which is commonly used in causal inference. Aiming to learn to rank with biased data with click propensities, Ai et al. [1] propose the Dual Learning Algorithm that combines an unbiased ranker and an unbiased propensity model. Bonner et al. [3] propose the CausE that is optimized with biased logged data and predicts recommendation results under random exposure. Considering that the missing rating in a recommendation system is usually missing not at random, Wang et al. [48] propose a doubly robust estimator for recommendation and further derive the tail bound of the estimator. Recently, Wang et al. [49] take the unexposed user-item pairs as the counterfactual samples, and propose the counterfactual variational information bottleneck. Chen et al. propose the autoDebias method, which leverages other uniform datasets to optimize the debiasing parameters. And the unbiased model-agnostic matching approach [26] leverages inverse propensity weighting to address the challenge that the negative samples from different sub-spaces at different stages have different importance. Motivated by the counterfactual propensity-weighting approach from causal inference, Xu et al. [51] address the unbiased recommendation problem by using a minimax empirical risk formulation. However, the aforementioned methods ignore that the historical logged data are collected under different strategies and these strategies are the reasons that lead to selection biases. Moreover, the aforementioned methods implicitly assume that the exposure strategies are stationary and this assumption is usually too strong. In this article, we address the selection biases problems in the recommendation by modeling the exposure strategies by combining the social networks with the causal generative process of rat level.

For social recommendation, one of the goals of recommendation is to learn better user variables, hence more and more researchers leverage the relationships among users with the consideration of homophily in the social network. Jamali et al. [15] combines matrix factorization with the mechanism of trust propagation of social networks to address the problems brought by cold-start users. Following the intuition that personal behaviors are affected by a person’s social network, Ma et al. [28] propose SoRec, which learns the user latent feature space and item latent feature space by employing the social networks and the user-item matrix simultaneously. To address the data sparsity and cold-start problem, Yang et al. [52] propose TrustMF, which employs the matrix factorization technique to map users into low-dimensional latent feature spaces in terms of their trust relationship. With the widespread use of deep learning, many researchers make use of neural networks to improve recommendation algorithms. Considering that the current recommendation largely relies on the initialization of the user and item latent feature vectors, Deng et al. [10] use deep learning to determine the initialization in the matrix factorization for the social recommendation. Considering that the users behave and interact differently in social networks and user-item bipartite graphs, Fan et al. [11] raise DASO, which adopts a bidirectional mapping method to transfer users’ information between the social domain and item domain. In this article, since both the user variables and the strategies variables are latent, it is hard to reconstruct them at the same time. Hence, we introduce the social
Table 1. Notations and Descriptions

| Notations | Descriptions |
|-----------|--------------|
| $u$       | User variables |
| $v$       | Item variables as well as the user variables |
| $e$, $r$  | Exposure variables and the rate-level variables |
| $s$       | Strategy variables |
| $\mathcal{U}$, $\mathcal{V}$ | User set and the item set |
| $E$, $R$  | Exposure matrix and rate-level matrix |
| $h$       | Features extracted by the models |
| $\mathcal{T}$, $\mathcal{O}$ | Exposed set and the unexposed set |
| $G$       | Social networks over $\mathcal{U}$ |
| $P(\cdot), Q(\cdot)$ | Probability distribution of random variables |
| $W_\ast, \Theta_\ast$ | Parameters of neural networks |
| $C(u)$    | Accessed items of user $u$ |
| $N(u)$    | First-order neighbors of $u$ |
| $f(\cdot), g(\cdot), \phi(\cdot)$ | Neural networks-based function |
| $\mathcal{F}_u$ | $\beta$-frequency neighbors item set, containing the items that have been accessed by at least $\beta$ neighbors of $u$ |
| $Z_{u,i}$ | Denotes the rank position of a positive feedback $(u, i)$ |

networks to reconstruct the user embedding first, then leverage it to reconstruct the strategies variables.

Other researchers borrow the idea of generative models. Zhou et al. [57] extend the flow-based generative model [32] to CF for modeling implicit feedback. And Liang et al. [22] combine multinomial likelihoods with collaborative filtering and extend variational auto-encoders [17] to collaborative filtering for implicit feedback. Liu et al. [25] consider both local and global structures among users under the Wasserstein auto-encoder frameworks. Recently, graph neural networks attract more and more attention, so some researchers combine generative models and graph neural networks. Yu et al. [53] propose a deep adversarial framework based on graph convolutional networks to address the problem of the sparsity of user-item relations and the noisy social relations. In this work, we bring the strategies variables into the structural causal model and tackle the recommendation problem as a counterfactual problem. We follow the paradigm of variational auto-encoders [17] to instantiate the proposed REST method.

3 IDENTIFICATION OF DEBIASED RECOMMENDATION

3.1 Notations

We first introduce the notations in this article. Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and items, respectively. We further let $E$ and $R$ denote the exposure matrix and the rate-level matrix defined over $\mathcal{U} \cup \mathcal{V}$. $e_{uv}$ is an element of $E$, with $e_{uv} = 1$ denotes that the item $v$ is exposed to the user $u$ and $e_{uv} = 0$ denotes that the item $v$ is not exposed to the $u$. $r_{uv}$ is an element of $R$, which denotes the rate level of $u$ on $v$. Hence, we let $\mathcal{T} = \{ < u, v, r_{uv} > | e_{uv} = 1 \}$ and $\mathcal{O} = \{ < u, v, r_{uv} > | e_{uv} = 0 \}$ be the exposed set and unexposed set, respectively. In the social recommendation context, a social network $G$ is associated with the user set $\mathcal{U}$. With the abuse of notation, we also let $u$, $v$ be the embedding of the corresponding entities and ignore the subscripts of $e_{uv}$ and $r_{uv}$. The mathematical notations used in this article are summarized in Table 1.
3.2 Causal Generation Process under Exposure Strategies and Social Networks

Based on the aforementioned notation description, we consider the causal graph to model the recommendation procedure. As shown in Figure 1(d), the causal graph contains six variables: $G, u, e, s, v$, and $r$. In particular, we let:

- $G \rightarrow u$ denote how social networks affect the interests of users,
- $u, v, s \rightarrow e$ denote that whether an item will be recommended depends on $u, v,$ and $s$,
- $u, e, v, s \rightarrow r$ denote that the exposure of item $v$ to user $u$ not only depends on $u$ and $v$ but also depends on the exposure strategies $s$.

Please note that our causal model (Figure 1(d)) is different from the existing debiased method (Figure 1(b)) from the following two aspects: (1) our model further takes the $s$ into consideration; (2) our model takes $u$ as latent variables and employs $G$ as the surrogate of $u$. This causal mechanism provides us a way to infer the latent variables $s$, because $G$ (similarly for $v$) and $s$ are dependent on each other conditioning on $e$. In other words, $G$ and $v$ provide us clues to infer the latent exposure strategies.

3.3 Social Recommendation as a Problem of Counterfactual Reasoning

Based on the aforementioned descriptions, we provide the definition of the social recommendation.

We first let $T'$ be the training set extracted from the exposed set, e.g., $T' \subseteq T$. Given the social networks $G$, the training set $T'$ and the strategy variables $s$, the goal of the social recommendation is to obtain a model that can estimate the following conditional distribution:

$$P(r|G, u, v, s, do(e = 1)), \tag{1}$$

in which $< u, v, r >$ is extracted from the unexposed set, i.e., $< u, v, r > \in O$. Note that the sample $< u, v, r >$ is extracted from the unexposed set but given $e = 1$, meaning that a user $u$ has never been exposed to an item $v$. And estimating the aforementioned conditional distribution equals answering the following question: What the rate level $r$ would be if an item $v$ is exposed to a user $u$ given exposure strategy $s$ and social networks $G$? Therefore, according to the theory of counterfactual inference [33], we can find that designing a social recommendation system is a counterfactual problem.

3.4 Identifying Unbiased Prediction of Social Recommendation System

Following the causal view of the recommendation systems, the goal of our social recommendation is to estimate the conditional distribution $P(r|G, u, v, do(e = 1))$ according to the do-calculus [33]. Note that the conditional distribution with the do-calculus is different from that forces the unexposed edges to “1.” In the recommendation case, given an unexposed user-item pair $(u, v)$, we need to estimate what if item $v$ was recommended to $u$, which is an intervention. And this intervention probability can be written as $P(v|G, v, do(e = 1))$. In the meanwhile, $P(r|G, u, v, s, e = 1)$ is only a conditional distribution from the observed data where $e = 1$. From the graphical view, do-calculus means the path $u \rightarrow e$ is blocked. The identification of such a counterfactual model is an immediate result of Pearl’s back-door criteria, as shown in Theorem 1.

**Theorem 1. (Identification of Social Recommendation)** Suppose that the joint distribution $P(G, e, r, v, s)$ is recovered, and the counterfactual prediction is identifiable under the causal model in Figure 1(d).

**Proof.** We prove that $P(r|G, v, do(e = 1))$ is identifiable under the premise of the theorem with the help of Equation (2):

---

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 2, Article 38. Publication date: November 2023.
where the second equality is based on the rule of do-calculus and conditional independent property under Figure 1(d) [33]. Essentially, we now can predict intervention based on the recovered joint distribution \( P(G, e, r, v, s) \), which finishes the proof.

\[\text{Proposition 1.} \] (KL-Divergence under Delta Distribution Assumption) KL-divergence \( D_{KL}(Q^*(u|G, e, r, v)||P(u)) \) is zero if \( P(u) \) is a delta distribution with the optimal parameters \( Q^* = \arg \max_Q \mathcal{L} \).

Please note that the aforementioned identification theorem of social recommendation shows that we can estimate the conditional distribution in Equation (1) with the help of social networks and the data extracted from the exposed set \( T \).

4 ALGORITHM AND IMPLEMENT

According to the causal graph shown in Figure 1(d), we devise a variational auto-encoders-based framework. We begin with the likelihood of the samples to derive the evidence lower bound (ELBO) of the model. Essentially, the logarithm of joint likelihood \( \ln P(G, e, r, v) \) can be written as follows:

\[
\ln P(G, e, r, v) = \mathcal{L}_{ELBO} + D_{KL}(Q(u|G, e, r, v)||P(u|G, e, r, v)) + D_{KL}(Q(s|e, r, v, u)||P(s|e, r, v, u))
\]

in which the second and third lines are the Kullback-Leibler divergence between the approximate distributions and the true posteriors. And \( \mathcal{L}_{ELBO} \) is the variational lower bound, which can be derived as follows (see more details in the Appendix):

\[
\mathcal{L}_{ELBO} = P(v) - D_{KL}(Q(s|e, r, v, u)||P(s)) + D_{KL}(Q(u|G, e, r, v)||P(u)) + \mathbb{E}_{Q(u|G, e, r, v)} \ln \left( D_{KL}(Q(s|e, r, v, u)||P(s|e, r, v, u)) \right)
\]

where \( Q(u|G, e, r, v) \) and \( Q(s|e, r, v, u) \) are the approximate functions that are also, respectively, named user latent variables encoder and the latent strategies variables encoder. These two encoders are used to approximate the two true posteriors: \( P(u|G, e, r, v) \) and \( P(s|e, r, v, u) \). And we further let \( P(G|u) \), \( P(e|v, u, s) \), and \( P(r|u, v, e) \) denote the social networks reconstruction, the exposure reconstruction, and the rating level prediction, respectively. To further facilitate the learning of the model, we assume that the latent user variables follow the delta distribution and the latent strategies variables \( s \) follow the categorical distribution. Since these two priors are also consistent with real-world recommendation systems.

Given the item variables \( v \), \( P(v) \) is a constant. Moreover, since we assume that \( P(u|G, e, r, v) \) is a delta distribution, the value of \( D_{KL}(Q(u|G, e, r, v)||P(u)) \) is equal to 0, the proof is provided in the Proposition 1.

\[\text{Proposition 1.} \] (KL-Divergence under Delta Distribution Assumption) KL-divergence \( D_{KL}(Q^*(u|G, e, r, v)||P(u)) \) is zero if \( P(u) \) is a delta distribution with the optimal parameters \( Q^* = \arg \max_Q \mathcal{L} \).
Fig. 2. Illustration of the framework of the proposed REST Model. (a) Inference phase contains the Latent User Variable Encoder and the Bias Encoder, which are used to reconstruct the user variables and the strategy variables, respectively. Note that KL loss denotes that we take the Kullback-Leibler divergence as the loss function. (b) Generation phase contains the rate-level reconstruction, the social network reconstruction, and the exposure reconstruction, which are used to, respectively, reconstruct the observed rating values, social structures, and the exposed variables. In the test step, we only employ the rate-level reconstruction for prediction.

Proof. We prove the aforementioned proposition by contradiction. First, we suppose that $D_{KL}(Q^*(u|G, e, r, v)||P(u)) \neq 0$. Then, given the delta distribution $P(u)$, there must exist an instance $u$ such that $Q^*(u|G, e, r, v) \neq 0, P(u) = 0$. It follows that $KL(Q^*(u|G, e, r, v)||P(u)) \rightarrow \infty$ leading to an under-optimized score $L \rightarrow -\infty$, which is a contradiction. □

Combining Proposition 1 and Equation (4), we can reformulate the objective function of the proposed REST model as follows:

$$L_t = D_{KL}(Q(s|e, r, v, u)||P(s))$$

$$- E_{Q(u|G, e, r, v)} \ln (P(G|u))$$

$$- E_{Q(u|G, e, r, v)} E_{Q(s|e, r, v, u)} \ln P(e|v, u, s)$$

$$- E_{Q(u|G, e, r, v)} \ln P(r|u, v, e),$$

(5)

in which $D_{KL}$ denotes the Kullback-Leibler divergence.

According to the objective function shown in Equation (5), we can find that the proposed model can be summarized into two phases: the inference phase and the generation phase, which are illustrated in Figure 2. Specifically, the inference phase, which is used to infer the latent variables, is composed of the user latent variable encoder $Q(u|G, e, r, v)$ and the latent strategies variables encoder $Q(s|e, r, v, u)$. The generation phase, which is used to infer the observational variables, is composed of the social network reconstruction $P(G|u)$, the rate-level reconstruction $P(r|u, v, e)$ and the exposure reconstruction $P(e|v, u, s)$. We will describe the implementation of the aforementioned components in the following subsections.

### 4.1 Inference Phase

#### 4.1.1 User Latent Variable Encoder.
In this part, we first introduce the details of the user latent variable encoder $Q(u|G, e, r, v)$ given the social network $G$, item $v$ as well as the corresponding rate level $r$, and exposure variables $e$. The procedure of inferring the user latent variables is composed of three steps. First, we aggregate the information of the bipartite graph to obtain the aggregated representation $h^B$. Second, we employ a similar way to obtain the aggregated representation $h^T$ on
social networks. Third, we split the aggregated representation for each type of exposure variable and then process them with different multilayer perceptrons (MLPs).

As for the first steps, we need to obtain the aggregated representation of the user-item bipartite graph, we employ the techniques of graph attention networks (GAT) [42]. In detail, given the user \( u_i \), the interacted item sets \( C(u) \) and the corresponding ratings, we obtain the aggregated representation \( h^b \) with the help of attention mechanism as follows:

\[
h^b = \sigma \left( \sum_{v \in C(u)} a^b_{uv} (v \oplus r) \right),
\]

\[
da^b_{uv} = \frac{g_b(u, v, r; W_b)}{\sum_{v' \in C(u)} g_b(u, v', r; W_b)},
\]

in which \( \oplus \) denotes the concatenation operation and \( g_b(\cdot) \) with trainable parameters \( W_b \) is the score function that is used to calculate the matching score given \( u, v, r \); \( a^b_{uv} \) denotes the important weights between user \( u \) and item \( v \). And \( \sigma(\cdot) \) denotes the LeakyReLU, which is the leaky version of a rectified linear unit; \( C(u) \) denotes the items list that \( u \) has accessed in the bipartite graph.

Second, we use another GAT to obtain the aggregated representation of social networks. Practically, we let \( G \) in \( Q(u|G, e, r, v) \) be the substructure of the social networks, for example, the first-order neighbors of \( u \). The calculation procedure is shown as follows:

\[
h^s = \sigma \left( \sum_{k \in N(u)} a^s_{uk} \cdot u \right),
\]

\[
da^s_{uk} = \frac{g_s(u_k; W_s)}{\sum_{u' \in N(u)} g_s(u, u', W_s)},
\]

in which \( g_s(\cdot) \) with trainable parameters \( W_s \) is the score function that is used to calculate the matching score given any two user embedding; \( a^s_{uk} \) denotes the important weight between \( u \) and \( k \); and \( N(u) \) denotes the first-order neighbors of \( u \).

Finally, to obtain the user latent variables, we devise the exposure-specific architecture inspired by CEVAE [27] and TARNet [36], which is shown in Figure 3. We use \( g_0(\cdot), g_1(\cdot) \) to generate the user latent variables \( h_u \). Specifically, we can obtain the latent user variables via the exposure-specific functions as follows:

\[
h_u = (1 - e) \ast g_0 \left( h^b, h^s; W^0_g \right) + e \ast g_1 \left( h^b, h^s; W^1_g \right),
\]

where \( g_0(\cdot) \) and \( g_1(\cdot) \) are composed of MLPs, \( W^0_g \) and \( W^1_g \) are the trainable parameters. For convenience, we let \( W_g = \{W^0_g, W^1_g, W_b, W_s\} \). For a tetrad \( (u, v, r, e = 1) \) in the exposed set, we use \( g_1(h^b \oplus h^s; W^1_g) \). For those from the unexposed set, we use \( g_0(h^b \oplus h^s; W^0_g) \).

The training of the unexposed \( g_0(\cdot) \) is crucial to the success of our counterfactual learning problem [27]. The main challenge is that we can only obtain the rate levels on the exposed set from the bipartite graph and the rate levels on the unexposed set are unavailable. To address this challenge,
we propose two different methods to approximate the unexposed set, which will be introduced as follows:

**Social Networks Voting-based Unexposed Set Estimation.** By assuming that the user behaviors are affected by the social networks, i.e., whether items will be chosen by users are totally decided by the behaviors of their friends, we first introduce the vote-based unexposed set estimation method, which can be separated into the following three steps:

- First, we extract $C(u)$ and $G_1^u$ for the user $u$, where $G_1^u$ is the set of 1-order neighbors of $u$.
- Second, we extract the $\beta$-frequency neighbors item set $\mathcal{F}_u$ by $\mathcal{F}_u = \{v \mid v \notin C(u), \sum_{u' \in N(u)} e_{u'v} \geq \beta\}$.
- Third, we obtain the unexposed sample $(u, v, r, e = 0)$, in which $v \in \mathcal{F}_u$ and $r$ is the most frequent rate level of $u$’s neighbors, i.e., using the voting method to get the value $r$ for the unexposed samples.

Please note that the aforementioned procedure to generate the counterfactual samples implicitly leverage the assumption that *both the users and their friends share similar interests and behaviors.*

**Pseudo-label-based Unexposed Set Estimation.** By assuming that the distribution of unexposed set is similar to the distribution of the exposed set, we propose the pseudo-label-based unexposed set estimation method. Moreover, since the user behaviors might be influenced by two factors, i.e., the social networks and the historical behaviors, we devise two different architectures to estimate the pseudo-labels.

First, we suppose that the behaviors of users are influenced by social networks, so we devise a social-networks-based unexposed set estimation method, which can be formalized as follows:

$$\hat{r} = f_p(G; W_G),$$

in which $f_p$ denotes the GCN-based architecture, $W_G$ denotes the parameters of $f_p$ and $\hat{r}_p$ denotes the pseudo labels.

Second, we suppose that the behaviors are influenced by historical behaviors, so we devise a user-behavior-based estimation method that uses the information of historical user behaviors, which can be formalized as follows:

$$\hat{r} = f_p(u_i, v_j),$$

in which $f_p$ denotes the MLP-based neural architecture and $v_j$ denotes the historical accessed item of $u_i$.

Finally, to optimize the aforementioned two architectures, we first train them with the data from the exposed set, then we can generate pseudo labels with these architectures.

**Random unexposed set estimation.** Since the user behaviors contain some randomness, we randomly assign a rate level for each unexposed item to make our model consider these randomnesses.

Note that we employ the vote-based unexposed set estimation method for the standard REST method, more experiments for different unexposed set estimation methods will be shown in the experiment section.

**4.1.2 Latent Strategies Variables Encoder.** In this subsection, we aim to model the latent strategies variables $s$ by using the exposure variables $e$, item variables $v$, rate level $r$, and latent user variables $u$. First, we follow the same aggregation method in Equation (6) to calculate the item aggregated representation $h^d$ for discrete strategies variables encoder, which is shown as
follows:

\[ h^d = \sigma \left( \sum_{v \in C(u)} a^d_{uv} (v \oplus r) \right), \]

\[ a^d_{uv} = \frac{g_d(u, v, r; W_d)}{\sum_{v \in C(u)} g_d(u, v', r; W_d)}, \]

in which \( W_d \) are the trainable parameters. Similar to Equation (8), we model the latent strategies variable with the help of another exposure-specific function as shown in Equation (12). Note that we use the Gumbel-Softmax trick \cite{16} to estimate latent strategies variable, since we assume they follow the categorical distribution:

\[ \hat{s} = (1 - e) \phi_0 \left( h^d, v; W^0_\phi \right) + e \phi_1 \left( h^d, v; W^1_\phi \right), \]

\[ s \sim \text{GUMBLE-SOFTMAX}(\hat{s}), \]

in which \( \phi_0 \) and \( \phi_1 \) are the exposure-specific function in latent strategies variables encoder. \( W^0_\phi \) and \( W^1_\phi \) are the trainable variables. For convenience, we let \( W_s = \{ W^0_\phi, W^1_\phi, W_d \} \).

### 4.2 Generation Phase

#### 4.2.1 Social Networks Reconstruction.

After obtaining the aforementioned two kinds of latent variables, we aim to reconstruct the social networks. In this article, we follow the configuration of variational graph auto-encoders \cite{18} and reconstruct each edge of social network structures of \( u \) as follows:

\[ \hat{G}_{u, u'} = \sigma (h_u, h_{u'}), \]

where \( \hat{G}_{u, u'} \) is the predicted edge between \( u \) and \( u' \). To train the model with the mini-batch, we only reconstruct the first-order neighbors of \( u \) instead of the whole social network.

#### 4.2.2 Exposure Reconstruction.

Given the latent user variables \( u \), latent strategies variables \( s \), and item variables \( v \), we aim to model \( P(e|v, u, s) \). We employ the following function to reconstruct the exposure variables:

\[ \hat{e} = f_e(h_u, v, s; \theta_e), \]

in which \( \theta_e \) are the trainable parameters and \( f_e(\cdot) \) is a neural architecture that is composed of MLPs.

#### 4.2.3 Rate Level Reconstruction.

Finally, we aim to predict the rate level, given the user latent variables \( u \), item variables \( v \), and exposure variables \( e \). Similar to Equation (8), we employ the exposure-specific rate level predictor, which is shown as follows:

\[ \hat{r}_{uv} = e \cdot f_1(h_u, v; \theta_1) + (1 - e) \cdot f_0(h_u, v; \theta_0), \]

in which \( f_0 \) and \( f_1 \) are also composed of MLPs and \( \theta_0 \) and \( \theta_1 \) are the trainable parameters. For convenience, we let \( \Theta_e = \{ \theta_0, \theta_1, \theta_e \} \).

### 4.3 Model Summarization

After combining the inference phase and the generation phase, we summarize the total loss of the proposed method as follows:

\[ \mathcal{L}(W_g, W_s, \Theta_e) = \mathcal{L}_t + \gamma \mathcal{L}_{reg}, \]

where \( \mathcal{L}_{reg} \) is the L2 regularization of the parameters; \( \gamma \) is the hyper-parameter.
Table 2. Statistics of the Datasets

| Dataset  | Epinions | Yelp | Ciao | WeChat |
|----------|----------|------|------|--------|
| # of users | 22K      | 332K | 7K   | 568K   |
| # of items | 296K     | 197K | 106K | 242K   |
| # of user-item relationship | 798K     | 4,567K | 282K | 9,422K |
| # of user-user relationship | 355K     | 7,043K | 57K  | 5,667K |
| Social networks density | 0.072%   | 0.0064% | 0.11% | 0.0018% |
| Bipartite graph density | 0.012%   | 0.0070% | 0.036% | 0.0068% |

During the training step, we optimize the model by using the following procedure:

\[
(W_t, W_s, \hat{\Theta}_r) = \arg\min_{W_t, W_s, \Theta_r} L(W_t, W_s, \Theta_r).
\] (17)

During the evaluation step, given the \( u \) and unexposed item \( v \), we let \( e = 1 \). The following procedure with the trained optimal parameters is adapted to the test dataset:

\[
\begin{align*}
\hat{h}_u &= g_t(h^b, h^s; W_g^1), \\
\hat{r} &= f_t(h_u, v; \theta_1).
\end{align*}
\] (18)

5 EXPERIMENT

In this section, we report experimental results on four datasets to evaluate our method against the state-of-the-art baselines, including the latest methods that use the idea of causal effect. With the help of the experiment results, we want to explore the following challenges: (1) Can the proposed REST method remove the disadvantage of selection biases? How is the performance compared with the existing methods? Especially the causality-based methods. (2) Can the strategies variables in the proposed REST method model effectively mitigate the non-stationary strategies challenges?

5.1 Datasets

To evaluate the performance of our method, we conduct experiments on three published datasets (including Ciao, Epinions, and Yelp) with explicit feedback and a private dataset collected from WeChat official accounts with implicit feedback. The details of the aforementioned dataset are shown in Table 2.

• Epinions\(^1\): A benchmark dataset for the social recommendation. In Epinions, a user can rate and give comments on items. Moreover, for each rating, this dataset provides the product name and its category, the rating score, the time point when the rating is created, and the helpfulness of this rating. Besides, a user can also select other users as their trustees. Note that we treat the trust graphs as social networks.

• Ciao\(^2\) is a published dataset for the social recommendation, which contains rating information of users given to items and also contains item category information. This dataset contains all the information Epinions has except the time points when the trust relations

\(^1\)http://www.trustlet.org/extended_epinions.html
\(^2\)www.ciao.co.uk
are established. The source cites of Ciao allows users to rate items, and add friends to their “Circle of Trust.”

- **Yelp**
  3: An online review platform where users review local businesses (e.g., restaurants and shops). This dataset is a subset of Yelp’s businesses, reviews, and user data for use in personal, educational, and academic purposes. The user-item interactions and the social networks are extracted in the same way as Epinions.

- **WeChat Official Accounts Dataset:** WeChat is a Chinese multi-purpose messaging, social media, and mobile payment application developed by Tencent. And WeChat official accounts dataset is one of the functions. On the WeChat Official Account platform, users can read and share articles. This dataset is constructed by user-article-clicking records and user-user social networks on this platform. We extract a subset of social networks and the bipartite graphs of users and articles as the whole dataset.

For each dataset, we follow Reference [5] and split the datasets chronologically. In detail, we split the datasets into four stages according to the interaction time and take the first two stages for the training set and the other two stages for the validation set and test set, respectively. We choose the model with the best validation and evaluate the chosen model on the test set. Note that we do not consider new users and new items in validation and testing. All the methods run with five different random seeds, and we report both the mean and variance. The source code and the preprocess scripts of the proposed methods are available at the following link.4

5.2 Hyper-parameters

We optimize all models with the Adam optimizer with a batch size of 1,024. For a fair comparison, all the methods are fine-tuned by searching the learning rate in the range of \{0.001, 0.0009, \ldots, 0.0001\}. We also adopt the early stopping strategy that stops training if RMSE/HR@20 on the validation dataset does not decrease/increase for 1,500 training steps.

5.3 Evaluation Metrics

We use different evaluation metrics for datasets with explicit feedback and implicit feedback, respectively.

For the dataset with explicit feedback, we use MSE and RMSE. The smaller values of MAE and RMSE, the better the predictive accuracy is. Note that even a small improvement in RMSE or MAE terms can have a significant impact on the quality of the top-few recommendations.

For the dataset with implicit feedback, we use Hit Rate@K (HR@K) and Normalized Discounted Cumulative Gain@K (NDCG@K). HR measures the percentage that recommended items contain at least one correct item interacted by the user, which can be formalized as follows:

\[
HR@K = \frac{|U_{hit}|}{|U_{all}|},
\]

in which \(|U_{hit}|\) is the number of users for which the correct answer is included in the top \(k\) recommendation list and \(|U_{all}|\) is the total number of users in the test dataset.

And NDCG takes the positions of correct recommended items into consideration, which measures the quality of recommendation through discounted importance based on position and

3https://www.kaggle.com/yelp-dataset/yelp-dataset
4https://github.com/DMIRLAB-Group/REST
can be formalized as follows:

$$\text{DCG}@k = \sum_{(u,i) \in U} I(\hat{Z}_{u,i} \leq k) \log(\hat{Z}_{u,i} + 1),$$

$$\text{NDCG}@k = \frac{1}{U} \sum_{u \in U} \frac{\text{DCG}_u@k}{\text{IDCG}_u@k},$$

in which $\hat{Z}_{u,i}$ denotes the rank position of a positive feedback $(u, i)$; $I(\cdot)$ denote the indicator function and $\text{IDCG}@k$ denotes the ideal $\text{DCG}_u@k$.

In this article, we choose $K$ in {5, 10, 20}. Note that higher scores of $\text{HR}@K$ and $\text{NDCG}@K$ indicate better performance.

### 5.4 Baselines

We compare the proposed REST method with four kinds of baselines. Besides the classical matrix factorization-based Methods, we also take some graph neural networks-based methods into account. Furthermore, we also compare our method with the baselines based on causal inference. Since our method uses the technique of variational influence, we also consider some VAE-based methods.

**Matrix Factorization-based Methods:**

- PMF [30]: Probabilistic Matrix Factorization is one of the most traditional methods for the recommendation that models latent factors of users and items by Gaussian distributions.
- NeuMF [14]: Neural network-based Collaborative Filtering replaces the inner product with a neural architecture that can learn an arbitrary function from data.
- BPRMF [34]: BPRMF, which is optimized by stochastic gradient descent with bootstrap sampling, is the maximum posterior estimator that is derived from the Bayesian theorem.

**Graph Neural Networks-based Methods:**

- GraphRec [12]: A graph neural networks-based method that leverages graph attention mechanism to aggregate the information of the social networks and user-item relations.
- LightGCN [13]: LightGCN optimizes the user and item representation by linearly propagating them on the bipartite graph, and uses the weighted sum of the representation.
- NGCF [47]: NGCF integrates the user-item interactions by modeling the high-order connectivity and injecting the collaborative signal into the embedding process.

**Variational Auto-encoder-based Methods:**

- MultVAE [22]: MultVAE extends VAE to collaborative filtering for implicit feedback, so it performs worse on the dataset with explicit feedback.
- RecVAE [38] uses the multinomial likelihood variational auto-encoders to map user feedback to user embeddings.

**Causality-based Methods:**

- CausE [3]: CausE jointly learns two CTR models and uses a multi-task objective that factorizes the matrix of observations.
- CVIB [49]: CVIB learns a balanced model based on Information Bottleneck, which simultaneously optimizes the factual and counterfactual embeddings. In this article, we compare our method with two variants of CVIB: MF-CVIB and NCF-CVIB.
### Table 3. Performance Evaluation of the Compared Methods on the Ciao, Epinion, and Yelp Dataset

| Model Class | Algorithms  | CIAO       |       | EPINIONS     |       | YELP       |       |
|-------------|-------------|------------|-------|--------------|-------|------------|-------|
|             |             | MAE        | RMSE  | MAE          | RMSE  | MAE        | RMSE  |
| MF          | PMF         | 0.9539±0.0040 | 1.1936±0.0019 | 1.0767±0.0035 | 1.2755±0.0022 | 0.9895±0.0023 | 1.2454±0.0011 |
|             | NeuMF       | 0.7770±0.0077 | 0.9828±0.0022 | 0.8457±0.0053 | 1.0838±0.0015 | 0.9575±0.0081 | 1.1958±0.0005 |
| VAE         | MultiVAE    | 0.9254±0.0025 | 1.1908±0.0014 | 0.9707±0.0010 | 1.2104±0.0039 | 0.9957±0.0031 | 1.2944±0.0020 |
|             | RecVAE      | 0.9449±0.0014 | 1.1787±0.0022 | 0.9614±0.0087 | 1.1964±0.0038 | 0.9944±0.0020 | 1.2385±0.0014 |
| Causality   | CausV      | 0.7943±0.0014 | 1.0003±0.0013 | 0.8553±0.0019 | 1.0705±0.0013 | 0.9400±0.0031 | 1.2039±0.0015 |
|             | CBIV-MF     | 0.9091±0.0016 | 1.2001±0.0011 | 0.9499±0.0031 | 1.2477±0.0003 | 0.9919±0.0012 | 1.3189±0.0024 |
|             | CBIV-NCF    | 0.7394±0.0027 | 1.0462±0.0013 | 0.8511±0.0128 | 1.2477±0.0003 | 0.9801±0.0011 | 1.3613±0.0043 |
|             | MACR-MF     | 0.9446±0.0051 | 1.1859±0.0030 | 0.9784±0.0092 | 1.2364±0.0031 | 0.9923±0.0004 | 1.2344±0.0004 |
|             | CIDR        | 0.8178±0.0045 | 1.0242±0.0012 | 0.8411±0.0021 | 1.0624±0.0012 | 0.9471±0.0032 | 1.2418±0.0016 |
|             | AutoDebias  | 0.7686±0.0042 | 1.0201±0.0020 | 0.8574±0.0038 | 1.0942±0.0019 | 0.9570±0.0027 | 1.2666±0.0021 |
|             | DecRS       | 0.7576±0.0038 | 0.9875±0.0033 | 0.8242±0.0043 | 1.0617±0.0033 | 0.9703±0.0000 | 1.2177±0.0007 |
| GNN         | GraphRec    | 0.7585±0.0051 | 0.9743±0.0021 | 0.8283±0.0019 | 1.0567±0.0019 | 0.9525±0.0035 | 1.1968±0.0017 |
|             | NGCF        | 0.8061±0.0023 | 1.0135±0.0010 | 0.9348±0.0023 | 1.1286±0.0017 | 0.9396±0.0023 | 1.2231±0.0017 |
|             | LightGCN    | 0.9373±0.0051 | 1.1919±0.0014 | 0.9584±0.0011 | 1.2025±0.0005 | 1.0015±0.0024 | 1.2444±0.0019 |
| Ours        | REST        | 0.7320±0.0117 | 0.9635±0.0009 | 0.8013±0.0045 | 1.0413±0.0007 | 0.9158±0.0054 | 1.1733±0.0006 |

The value presented is averaged over 5 replicated with different random seeds. The standard deviation is in the subscript.

- MACR-MF [50]: MACR-MF leverages the idea of causal effect and builds a multi-task learning schema over MF. We compare MACR-MF with our method in the implicit feedback dataset.
- DecRS [43]: Deconfounded Recommender System (DecRS) models the causal effect of user representation on the prediction score, which eliminates the impact of the confounder with the help of backdoor adjustment. Note that we only compare DecRs in the Ciao and Epinion datasets, since this method needs the categories of items and the Yelp dataset does not contain the item categories.
- CIDR [24] use the biased observational data to capture a substitute of confounders on both the user side and item side and boost counterfactual inference to remove the causal effect of such confounders for better recommendation performance.
- AutoDebias [7] analyze the origin of biases from the perspective of risk discrepancy and optimize the debiasing parameters by solving the bi-level optimization problem with the help of meta-learning.

### 5.5 Deconfounding Performance

In this subsection, we aim to answer the following question: (1) Can the proposed REST method remove the disadvantage of selection biases? (2) And how is the performance compared with the existing methods, including the latest causality-based method?

#### 5.5.1 Experiment Results on Datasets with Explicit Feedback

We first illustrate the experimental results on the explicit feedback dataset, in which the users provide the rating for items. Hence, we follow Reference [12] and employ the MAE and RMSE as the evaluation metric. The experiment results on Ciao, Epinions, and Yelp datasets are shown in Table 3. From the experiment results, we can obtain the following observations:

- The proposed REST method outperforms the other methods by a large margin, which proves that our method can effectively remove the disadvantages of the selection biases. Furthermore, the superior performance of the proposed REST reflects the advantages of the identification theorem.
- According to Table 3, the proposed REST achieves different degrees of improvement on the three explicit datasets. In detail, the REST, respectively, obtains 26.4%, 18.1%, and 7.7% improvements on the Ciao, Epinion, and Yelp datasets. This is because the social network

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 2, Article 38. Publication date: November 2023.
### Table 4. Performance Evaluation of the Compared Methods on the WeChat Dataset

| Model Class | Models          | HR@5        | NDCG@5      | HR@10       | NDCG@10     | HR@20       | NDCG@20     |
|-------------|----------------|-------------|-------------|-------------|-------------|-------------|-------------|
| MF          | BPRMF          | 56.16±0.16  | 44.43±0.14  | 67.11±0.13  | 47.98±0.14  | 77.47±0.16  | 50.60±0.13  |
|             | NeuMF          | 60.76±0.33  | 48.20±0.45  | 71.18±0.32  | 51.57±0.42  | 81.27±0.59  | 54.13±0.49  |
| VAE         | SocialMF       | 41.78±0.16  | 32.57±0.29  | 50.01±0.18  | 35.23±0.29  | 58.23±0.14  | 37.31±0.25  |
|             | MultiVAE       | 53.15±0.07  | 41.54±0.06  | 64.97±0.08  | 45.36±0.07  | 76.59±0.08  | 48.30±0.07  |
|             | RecVAE         | 55.54±0.08  | 43.55±0.06  | 67.34±0.09  | 47.37±0.06  | 78.64±0.06  | 50.25±0.05  |
| GNN         | GraphRec       | 54.61±1.63  | 42.47±1.29  | 66.97±1.74  | 46.35±1.32  | 77.77±1.58  | 49.18±1.28  |
|             | LightGCN       |             |             |             |             |             |             |
|             | NGCF           |             |             |             |             |             |             |
| Causality   | MF-CVIB        | 62.28±0.26  | 50.14±0.27  | 72.40±0.21  | 53.43±0.25  | 81.49±0.18  | 55.73±0.24  |
|             | NCF-CVIB       | 63.89±0.70  | 52.27±0.98  | 72.84±0.49  | 55.17±0.91  | 81.18±0.46  | 57.28±0.87  |
|             | MACR-MF        | 61.81±0.07  | 48.64±0.23  | 72.34±0.30  | 52.06±0.11  | 81.23±0.46  | 54.32±0.07  |
|             | CausE          | 59.54±0.31  | 41.54±0.06  | 64.97±0.08  | 45.36±0.07  | 76.59±0.08  | 48.30±0.07  |
| Ours        | REST           | 65.31±0.27  | 52.07±0.19  | 76.05±0.20  | 55.56±0.19  | 85.37±0.13  | 57.92±0.14  |

The value presented is averaged over 5 replicated with different random seeds. The standard deviation is in the subscript.

- Densities of these datasets are different. According to Table 2, we can find that the Ciao dataset contains the densest social networks while the Yelp contains the sparsest one. This is because denser social networks can provide more counterfactual samples, which further benefit the model performance.
- Our method not only outperforms the conventional recommendation algorithms like PMF and NeuralMF but also outperforms the VAE-based methods like MultiVAE and RecVAE. This is because the VAE-based methods assume that the distributions of latent variables follow the Gaussian distribution but the assumption is too strong and does not work in practice. In the meanwhile, assuming that the latent variables follow the delta distribution, the proposed REST method can avoid the aforementioned drawback.
- The graph neural-network-based methods like the LightGCN and the GraphRec, which are designed for the implicit feedback datasets, perform poorly in the explicit feedback dataset. For one thing, this verifies that the graph neural networks are still poisoned by selection biases even though they leverage social networks. For another thing, the proposed method leverage the social to generate counterfactual samples that can mitigate the selection biases to some extent.
- As for the causal inference-based method, our method outperforms the causality-based method like CausE, MACR-MF, and DecFM. This is because the proposed REST method models latent strategy variables that break the stationary strategy assumption. We will further explore the effectiveness of the latent strategy variables in the following subsections.

#### 5.5.2 Experiment Results on the Dataset with Implicit Feedback

Then, we further illustrate the experimental results on the implicit feedback dataset, in which only the actions of users like clicking or purchasing, are collected. Hence, we follow Reference [50] and employ HR@K and NDCG@K as the evaluation metrics. The implicit feedback scenario is more challenging, because it is hard to distinguish if the unseen samples are disliked or not. The experiment results on the WeChat Official Account dataset are shown in Table 4. We do not report the experiment result of LightGCN and NGCF because of the limited CPU memory. According to the experiment results, we can get the following conclusions:

- As similar to the experiment results on the explicit feedback dataset, we can find that the proposed REST method still achieves the best performance, which reflects that our method can work on both the explicit and the implicit scenarios.
Table 5. Loss Weight Sensitivity of Reconstructing $r$ on Ciao and Epinions

| Model   | Ciao         | Epinions       |
|---------|--------------|----------------|
|         | MSE          | RMSE           | MSE          | RMSE           |
| $w_r = 0.1$ | 0.7520±0.0031 | 0.9825±0.0044 | 0.8080±0.0126 | 1.0468±0.0017 |
| $w_r = 0.5$ | 0.7507±0.0083 | 0.9717±0.0035 | 0.8062±0.0018 | 1.0429±0.0025 |
| $w_r = 1.0$ | 0.7320±0.0117 | 0.9635±0.0009 | 0.8013±0.0045 | 1.0413±0.0007 |
| $w_r = 5.0$ | 0.7386±0.0099 | 0.9683±0.0026 | 0.7981±0.0089 | 1.0427±0.0007 |
| $w_r = 10.0$ | 0.7443±0.0041 | 0.9688±0.0020 | 0.8010±0.0062 | 1.0418±0.0005 |

Table 6. Loss Weight Sensitivity of Reconstructing $G$ on Ciao and Epinions

| Model   | Ciao         | Epinions       |
|---------|--------------|----------------|
|         | MSE          | RMSE           | MSE          | RMSE           |
| $w_G = 0.1$ | 0.7320±0.0117 | 0.9635±0.0009 | 0.8013±0.0045 | 1.0413±0.0007 |
| $w_G = 0.5$ | 0.7470±0.0054 | 0.9701±0.0024 | 0.7967±0.0072 | 1.0419±0.0006 |
| $w_G = 1.0$ | 0.7472±0.0056 | 0.9700±0.0028 | 0.8063±0.0015 | 1.0434±0.0016 |
| $w_G = 5.0$ | 0.7447±0.0069 | 0.9733±0.0047 | 0.7984±0.0084 | 1.0427±0.0006 |
| $w_G = 10.0$ | 0.7485±0.0087 | 0.9776±0.0044 | 0.8068±0.0035 | 1.0455±0.0001 |

- Compared with the causal-based methods like MACR-MF [50] and the other types of methods, the causality-based methods achieve a better result, which reflects that the selection biases really harm the performance and taking causality into consideration will ease the disadvantage to some extent.
- In the meanwhile, our method also achieves good results. This is because the WeChat dataset is more likely controlled by different types of strategies like different fast-breaking news. Therefore, taking the stationary assumption and ignoring the strategies will degenerate the performance of the recommendation systems even the selection biases have been taken into account.

5.6 Ablation Analysis

To evaluate the effectiveness of the latent unobserved strategy variables, we raise a model invariant named REST-S, which removes the latent unobserved strategy variables in the data generation process. In this case, we follow the stationary strategy assumption and do not model the strategies behind the data.

5.6.1 Study of Importance of Individual Loss of Different Reconstruction. We further investigate the importance of the individual loss of reconstructing $r$, $G$, and $e$ on the final prediction, which are shown in Tables 5, 6, and 7, respectively. According to the experiment results, we can observe we can achieve the best performance with the default hyperparameters.

5.6.2 Effectiveness of the Discrete Strategy Variables. To verify the effectiveness of the discrete exposure component of our model, we devise REST-S. The experiment results are shown in Figures 4 and 5. Based on the experiment results, we can observe that:

- Compared REST-S with the standard REST, we can find that the performance of REST-S is lower than that of REST, which reflects the advantages of modeling the discrete strategies.
- Since we do not model the discrete strategies in REST-S, both the REST-S and other causality-based methods like MACR-MF are the same from the view of principle, so it is reasonable to guess that the performance of both the REST-S and other causality-based methods are similar.
Table 7. Loss Weight Sensitivity of Reconstructing $e$ on Ciao and Epinions

| Model | Ciao | | | Epinions | | |
|-------|-----|-----|-----|----------|-----|
|       | MSE | RMSE | MSE | RMSE | MSE | RMSE |
| $w_e = 0.1$ | 0.7359 ± 0.0042 | 0.9694 ± 0.0012 | 0.7997 ± 0.0040 | 1.0410 ± 0.0011 |
| $w_e = 0.5$ | 0.7384 ± 0.0055 | 0.9705 ± 0.0023 | 0.7955 ± 0.0067 | 1.0437 ± 0.0011 |
| $w_e = 1.0$ | 0.7320 ± 0.0117 | 0.9635 ± 0.0009 | 0.8013 ± 0.0045 | 1.0413 ± 0.0007 |
| $w_e = 5.0$ | 0.7530 ± 0.0098 | 0.9725 ± 0.0045 | 0.8015 ± 0.0067 | 1.0405 ± 0.0003 |
| $w_e = 10.0$ | 0.7413 ± 0.0027 | 0.9744 ± 0.0007 | 0.8138 ± 0.0069 | 1.0425 ± 0.0013 |

Compared REST-S with the other baselines, like CVIB-NCF and MACR-MF, we can find that we still obtain a comparable performance, which not only validates the aforementioned guess but also the effectiveness of modeling strategies.

5.6.3 Effectiveness of the Social Networks. To evaluate the effectiveness of the social, we apply the proposed method on social networks with different sparsity, which is named REST-sp-x (x denotes the sparsity of social networks, i.e., REST-sp-25 denotes that we apply social networks with 25% edges to the proposed REST method). Based on the experiment results on Ciao, Epinions, and Yelp datasets shown in Table 8, we can observe that:

- Compared the proposed REST with that on the sparse social networks, the sparser the social networks are, the better experiment results the proposed method can achieve, reflecting that the social networks benefit recommendation performance.
- Compared the REST on sparse social networks with other compared methods, our method still outperforms most of the baselines, reflecting that the proposed method can leverage the exposure strategies and achieve better performance.
Table 8. Performance of the Proposed REST Method on Social Networks with Different Sparsity

| Model          | Ciao MSE | Ciao RMSE | Epinions MSE | Epinions RMSE | Yelp MSE | Yelp RMSE |
|---------------|---------|----------|--------------|---------------|---------|-----------|
| REST-sp-0     | 0.7427±0.0105 | 0.9824±0.0018 | 0.8134±0.0037 | 1.0609±0.0014 | 0.9370±0.0083 | 1.1978±0.0010 |
| REST-sp-25    | 0.7357±0.0102 | 0.9674±0.0018 | 0.8034±0.0043 | 1.0469±0.0015 | 0.9172±0.0061 | 1.1776±0.0008 |
| REST-sp-50    | 0.7401±0.0098 | 0.9754±0.0013 | 0.8090±0.0040 | 1.0578±0.0012 | 0.9223±0.0048 | 1.1892±0.0009 |
| REST-sp-75    | 0.7412±0.0096 | 0.9797±0.0016 | 0.8109±0.0032 | 1.0580±0.0011 | 0.9353±0.0039 | 1.1951±0.0015 |
| REST          | **0.7320±0.0117** | **0.9635±0.0009** | **0.8013±0.0045** | **1.0413±0.0007** | **0.9158±0.0054** | **1.1733±0.0006** |

Table 9. Experiment Results on Several Unexposed Set Estimation Methods on Ciao and Epinion Datasets

| Model                                      | Ciao MSE | Ciao RMSE | Epinions MSE | Epinions RMSE | Yelp MSE | Yelp RMSE |
|--------------------------------------------|---------|----------|--------------|---------------|---------|-----------|
| Social Networks Voting-based estimation    | 0.7320±0.0017 | 0.9639±0.0009 | 0.8013±0.0045 | 1.0413±0.0007 |
| Social-network-based pseudo-label Estimation | 0.7392±0.0043 | 0.9626±0.0016 | 0.8040±0.0080 | 1.0414±0.0004 |
| User-behavior-based Pseudo-label Estimation | 0.7358±0.0049 | 0.9617±0.0019 | 0.8033±0.0067 | 1.0417±0.0010 |
| Random Estimation                          | 0.7411±0.0036 | 0.9649±0.0008 | 0.8075±0.0026 | 1.0414±0.0007 |
| Weighted-Average Ensemble Strategy         | 0.7336±0.0045 | 0.9629±0.0010 | 0.8088±0.0067 | 1.0419±0.0012 |
| Voting Ensemble Strategy                   | 0.7319±0.0052 | 0.9642±0.0026 | 0.8015±0.0061 | 1.0410±0.0007 |

5.6.4 Effectiveness of Different Unexposed Set Estimation Methods. To evaluate the performance of different unexposed set estimation methods, we not only evaluate the performance of our REST method on different unexposed set estimation methods but also explore the effectiveness of these unexposed set estimation methods with two different ensemble strategies, i.e., the weighted average ensemble strategy and the voting ensemble strategy. Experiment results on the Ciao and Epinion datasets can be shown in Table 9. According to the experiment results, we can observe that:

- Compared the experiment results of different unexposed set estimation methods, we can find that the performance of the social network voting-based estimation method is better than two types of pseudo-based estimation method, showing that the distribution between the exposed and unexposed dataset are different. We also find that the performance of the random estimation methods is worse than other estimation methods, this is reasonable, since the randomness strategy brings some noise.
- Compared the experiment results on different ensemble strategies, we can find that both the ensemble strategies achieve comparable performance on the Ciao dataset and better performance on the Epinions dataset, meaning that the ensemble strategies can involve the advantages of different unexposed estimation methods. Moreover, we also find that the performance of the weighted-average ensemble strategy is better than that of the voting ensemble strategy, this is because the voting ensemble strategy might easily ignore the results of the random estimation method and reduce the generalization of the model.

We further consider two types of cross-domain methods, i.e., DANN and MMD, to explore the different distributions of the exposed and unexposed datasets. In detail, we first employ the ensemble strategy to generate the unexplored set. Then, we use the adversarial technique like DANN and distribution minimizing restrictions like MMD to align the distributions of user and item variables. Experiment results are shown in Table 10. According to the experiment results, we can find that the performance of DANN and MMD is lower than that of the ensemble strategy, this is because some domain-specific semantic information is lost when the cross-domain methods extract the domain-invariant information.
Table 10. Experiment Results on Ciao and Epinion Dataset with Cross-domain Methods to Address the Distribution Shift Challenges between the Exposed and Unexposed Datasets

| Model | Ciao     | Epinions |
|-------|----------|----------|
|       | MSE      | RMSE     | MSE      | RMSE     |
| DANN  | 0.7446±0.0040 | 0.9700±0.0020 | 0.8016±0.0016 | 1.0405±0.0006 |
| MMD   | 0.7563±0.0089 | 0.9819±0.0043 | 0.8251±0.0017 | 1.0582±0.0025 |

Fig. 6. Visualization of the discrete strategies variables. The vertical coordinate and the horizontal coordinate denote the different festivals and the different dimensions in the form of the one-hot vector. (*Best viewed in color.*)

5.7 Visualization

To further show the necessity for modeling the latent discrete strategies, we provide the visualization of the discrete strategies with 64 dimensions on the Ciao dataset, which is shown in Figure 6. We split the 64 dimensions into 16 different categorical distributions, which represent 16 different one-hot vectors. Note that the horizontal axis stands for each dimension of latent variables. We choose three different traditional festivals, Christmas, Thanksgiving Day, and Valentine’s Day, and draw the latent discrete strategies variables of the same user at each festival in different years. The value of the yellow block is 1 and the value of the purple is 0. According to the visualization, we can find that:

- Shared patterns among the reconstructed strategy variables come from the same festival. For example, on Christmas, the locations of the yellow blocks are similar. This means that the same festival shares similar promotion strategies.
- The latent discrete strategies variables from different festivals look different, which means that different festivals have different promotion strategies. By modeling the strategies variables, we can well model the complex user-item relationships despite the disadvantages of selection bias.

6 CONCLUSION

This article presents a debiased recommendation framework based on explicitly modeling and reconstructing the discrete unobserved exposure strategies. In the proposed method, we reconstruct the latent exposure variables from the observational data using a variational auto-encoder framework, with the help of the clues from both the social networks and the items. The correctness and effectiveness of our proposal are verified on four real-world datasets. The success of our model not only reveals that the latent exposure strategies are the cause of the well-known selection bias problem but also provides an effective solution for this open problem in the recommendation...
The visualization of the recovered exposure strategies on the real-world dataset also provides some interesting insights into the existing recommendation systems.

A APPENDIX

The proofs of evidence lower bound (ELBO):

\[
\ln P(G, e, r, v) \geq P(v) - D_{KL}(Q(s|e, r, v, u || P(s)) \\
- D_{KL}(Q(u|G, e, r, v) || P(u)) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \ln (P(G|u)) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \mathbb{E}_{Q(s|e, r, v, u)} \ln P(e|v, u, s) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \ln P(r|u, v, e).
\]

**Proof.** The proof of the ELBO is composed of three steps. First, we factorize the conditional distribution according to the Bayes theorem:

\[
\ln P(G, e, r, v) = \ln \frac{P(G, e, r, v, u, s)}{P(u, s|G, e, r, v)} = \ln \frac{P(G, e, r, v, u, s)Q(u|G, e, r, v)Q(s|e, r, v, u)}{P(u|G, e, r, v)P(s|e, r, v, u)Q(u|G, e, r, v)Q(s|e, r, v, u)}.
\]

Second, we add the expectation operator on both sides of the equation and reformalize the equation as follows:

\[
\ln P(G, e, r, v) = D_{KL}(Q(u|G, e, r, v) || P(u|G, e, r, v)) \\
+ D_{KL}(Q(s|e, r, v, u) || P(s|e, r, v, u)) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \mathbb{E}_{Q(s|e, r, v, u)} \ln \frac{P(G, e, r, v, u, s)}{Q(u|G, e, r, v)Q(s|e, r, v, u)}.
\]

Third, we obtain the last equality with the help of \(D_{KL}(|\cdot|) \geq 0\):

\[
\ln P(G, e, r, v) \geq \mathbb{E}_{Q(u|G, e, r, v)} \mathbb{E}_{Q(s|e, r, v, u)} \ln \frac{P(G, e, r, v, u, s)}{Q(u|G, e, r, v)Q(s|e, r, v, u)} = \mathbb{E}_{Q(u|G, e, r, v)} \mathbb{E}_{Q(s|e, r, v, u)} \ln \frac{P(u)P(G|u)P(e|v, u, s)P(r|u, v, e, s)}{Q(u|G, e, r, v)Q(s|e, r, v, u)} \\
= - D_{KL}(Q(u|G, e, r, v) || P(u)) - D_{KL}(Q(s|e, r, v, u) || P(s)) + P(v) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \ln (P(G|u)) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \mathbb{E}_{Q(s|e, r, v, u)} \ln P(e|v, u, s) \\
+ \mathbb{E}_{Q(u|G, e, r, v)} \ln P(r|u, v, e, s).
\]

\[\square\]

**ACKNOWLEDGMENTS**

We thank Lingling Yi and Li Li from WeChat for their help and support on this work.
REFERENCES

[1] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity score estimation. In Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. 385–394.

[2] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza. 2004. Query recommendation using query logs in search engines. In Proceedings of the International Conference on Extending Database Technology. Springer, 588–596.

[3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In Proceedings of the 12th ACM Conference on Recommender Systems. 104–112.

[4] Sarah Bouraga, Ivan Jureta, Stéphane Faulkner, and Caroline Herssens. 2014. Knowledge-based recommendation systems: A survey. Int. J. Infell. Info. Technol. 10, 2 (2014), 1–19.

[5] Rocio Cañamares and Pablo Castells. 2018. Should I follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR’18). ACM, New York, NY, 415–424. https://doi.org/10.1145/3209978.3210014

[6] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. 2020. Controllable multi-interest framework for recommendation. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2942–2951.

[7] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. AutoDebias: Learning to debias for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’21). ACM, New York, NY, 21–30. https://doi.org/10.1145/3404835.3462919

[8] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. Retrieved from https://arXiv:2010.03240

[9] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems. 191–198.

[10] Shuiguang Deng, Longtao Huang, Guandong Xu, Xindong Wu, and Zhaohui Wu. 2016. On deep learning for trust-aware recommendations in social networks. IEEE Trans. Neural Netw. Learn. Syst. 28, 5 (2016), 1164–1177.

[11] Wenqi Fan, Tyler Derr, Yao Ma, Jianping Wang, Jiliang Tang, and Qing Li. 2019. Deep adversarial social recommendation. Retrieved from https://arXiv:1905.13160

[12] Wenqi Fan, Yao Ma, Qiang Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In Proceedings of the World Wide Web Conference. 417–426.

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 639–648.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th International Conference on World Wide Web. 173–182.

[15] Moshen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation. In Proceedings of the 15th ACM Conference on Recommender Systems. 191–198.

[16] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. Retrieved from https://arXiv:1611.01144.

[17] Diederik P. Kingma and Max Welling. 2013. Auto-encoding variational bayes. Retrieved from https://arXiv:1312.6114

[18] Thomas N. Kipf and Max Welling. 2016. Variational graph auto-encoders. Retrieved from https://arXiv:1611.07308

[19] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrixfactorizationtechniquesforrecommendersystems. In Proceedings of the 1st ACM Conference on Recommender Systems. 417–426.

[20] Zijian Li, Ruichu Cai, Fengzhu Wu, Sili Zhang, Hao Gu, Yuexing Hao, and Yuguang Yan. 2022. TEA: A sequential recommendation framework via temporally evolving aggregations. IEEE Transactions on Neural Networks and Learning Systems (2022), 1–12. https://doi.org/10.1109/TNNLS.2022.3

[21] Dawen Liang, Laurent Charlin, and David M. Blei. 2016. Causal inference for recommendation. In Proceedings of the Causation: Foundation to Application Workshop at UAI (AUAI’16).

[22] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tomy Jebara. 2018. Variational autoencoders for collaborative filtering. In Proceedings of the World Wide Web Conference. 689–698.

[23] G. Linden, B. Smith, and J. York. 2003. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput. 7, 1 (2003), 76–80. https://doi.org/10.1109/MIC.2003.1167344

[24] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiugiang He, Wei Ke Pan, and Zhong Ming. 2021. Mitigating confounding bias in recommendation via information bottleneck. In Proceedings of the 15th ACM Conference on Recommender Systems (RecSys’21). ACM, New York, NY, 351–360. https://doi.org/10.1145/3460231.3474263

[25] Huafeng Liu, Liping Jing, Jingxuan Wen, Zicheng Wu, Xiaoyi Sun, Jiaqi Wang, Lin Xiao, and Jian Yu. 2020. Deep global and local generative model for recommendation. In Proceedings of the Web Conference. 551–561.

ACM Transactions on Knowledge Discovery from Data, Vol. 18, No. 2, Article 38. Publication date: November 2023.
[26] Jiazhen Lou, Hong Wen, Fuyu Lv, Jing Zhang, Tengfei Yuan, and Zhao Li. 2022. Re-weighting negative samples for model-agnostic matching. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR ’22). ACM, New York, NY, 1823–1827. https://doi.org/10.1145/3477495.3532053

[27] Christos Louizos, Uri Shalit, Joris Mooij, David Sontag, Richard Zemel, and Max Welling. 2017. Causal effect inference with deep latent-variable models. In Proceedings of the 31st International Conference on Neural Information Processing Systems. 6449–6459.

[28] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. 2008. Sorec: Social recommendation using probabilistic matrix factorization. In Proceedings of the 17th ACM Conference on Information and Knowledge Management. 931–940.

[29] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. Retrieved from https://arXiv:1910.14238

[30] Andriy Mnih and Russ R. Salakhutdinov. 2007. Probabilistic matrix factorization. Adv. Neural Info. Process. Syst. 20 (2007), 1257–1264.

[31] Zohreh Ovaisi, Rajib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for selection bias in learning-to-rank systems. In Proceedings of the Web Conference. 1863–1873.

[32] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. 2019. Normalizing flows for probabilistic modeling and inference. Retrieved from https://arXiv:1912.02762

[33] Judea Pearl. 2009. Causality. Cambridge University Press.

[34] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In Proceedings of the International Conference on Machine Learning. PMLR, 1670–1679.

[35] Uri Shalit, Fredrik D. Johansson, and David Sontag. 2017. Estimating individual treatment effect: Generalization bounds and algorithms. In Proceedings of the International Conference on Machine Learning. PMLR, 3076–3085.

[36] Amit Sharma, Jake M. Hofman, and Duncan J. Watts. 2015. Estimating the causal impact of recommendation systems from observational data. In Proceedings of the 16th ACM Conference on Economics and Computation. 453–470.

[37] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I. Nikolenko. 2020. RecVAE: A new variational autoencoder for Top-N recommendations with implicit feedback. In Proceedings of the 13th International Conference on Web Search and Data Mining. 528–536.

[38] Chunyao Song, Yao Ge, Tingjian Ge, Haixia Wu, Zhutian Lin, Hong Kang, and Xiaojie Yuan. 2021. Similar but foreign: Link recommendation across communities. Info. Sci. 552 (2021), 142–166.

[39] Harald Steck. 2011. Item popularity and recommendation accuracy. In Proceedings of the 5th ACM Conference on Recommender Systems. 125–132.

[40] Moshe Tennenholtz and Oren Etzioni. 2019. Rethinking search engines and recommendation systems: A game theoretic perspective. Commun. ACM 62, 12 (2019), 66–75.

[41] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. Retrieved from https://arXiv:1710.10903

[42] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2020. “Click” is not equal to “like”: Counterfactual recommendation for mitigating clickbait issue. Retrieved from https://arXiv:2009.09945

[43] Xuanchui Wang, Michael Bendersky, Donald Metzler, and Marc Najork. 2016. Learning to rank with selection bias in personal search. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval. 115–124.

[44] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 165–174.

[45] Xingyi Wei, Fuli Feng, Jiawei Chen, Chufeng Shi, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2019. Doubly robust joint learning for recommendation on data missing not at random. In Proceedings of the International Conference on Machine Learning. PMLR, 6638–6647.

[46] Zifeng Wang, Xi Chen, Rui Wen, Shao-Lun Huang, Ercan E. Kuruglu, and Yefeng Zheng. 2020. Information theoretic counterfactual learning from missing-not-at-random feedback. Retrieved from https://arXiv:2009.02623
[51] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Adversarial counterfactual learning and evaluation for recommender system. *Adv. Neural Info. Process. Syst.* 33 (2020).

[52] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2017. Social collaborative filtering by trust. *IEEE Trans. Pattern Anal. Mach. Intell.* 39, 8 (2017), 1633–1647.

[53] Junliang Yu, Hongzhi Yin, Jundong Li, Min Gao, Zi Huang, and Lizhen Cui. 2022. Enhancing social recommendation with adversarial graph convolutional networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2022), 3727–3739. https://doi.org/10.1109/TKDE.2020.303

[54] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Comput. Surveys* 52, 1 (2019), 1–38.

[55] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Found. Trends Info. Retr.* 14, 1 (2020), 1–101.

[56] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. Retrieved from https://arXiv:2105.06067

[57] Fan Zhou, Yuhua Mo, Goce Trajcevski, Kunpeng Zhang, Jin Wu, and Ting Zhong. 2020. Recommendation via collaborative autoregressive flows. *Neural Netw.* 126 (2020), 52–64.

Received 10 May 2022; revised 24 March 2023; accepted 6 September 2023