Causal Inference for Multi-Criteria Rating Recommender Systems

ZHIHAO GUO, Shanxi University, China
PENG SONG*, Shanxi University, China
CHENJIAO FENG, Shanxi University of Finance and Economics, China
KAIXUAN YAO, Shanxi University, China
JIYE LIANG, Shanxi University, China

Recommender systems are designed to assist users in discovering interesting items and bringing profits to online platforms. The existing works primarily explore the correlation between historical feedback and model predictions through the data-driven paradigm based on a single user-item rating matrix (i.e., overall rating). However, this single-criterion methods ignore the users' multi-criteria (MC) behavioral characteristics. For example, a hotel system allows users to rate from multiple dimensions, such as environment and location (i.e., MC ratings). Moreover, selection bias is pervasive in user behavior data. Traditional data-driven methods may induce spurious association and amplified biases. To address the above challenges, we propose a debiasing framework called *Multi-Criteria Causal Recommendation* (MCCR), which encapsulates users' diverse MC preferences and employs causal inference to construct novel training and inference strategies. Specifically, we first represent the causal relationships among variables in MC scenarios through the structural causal model. Then, we mitigate the negative impact of selection bias through the back-door adjustment. Next, a graph representation learning framework suitable for MC ratings is developed, which is used to extract higher-order information and infer the heterogeneity of users' preferences with different criteria. Experimental results on six real datasets demonstrate that the MCCR significantly outperforms the existing baselines.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**; • **Information systems** → **Recommender systems**.

Additional Key Words and Phrases: Recommender Systems, Multi-Criteria Recommendation, Causal Inference, Debiasing

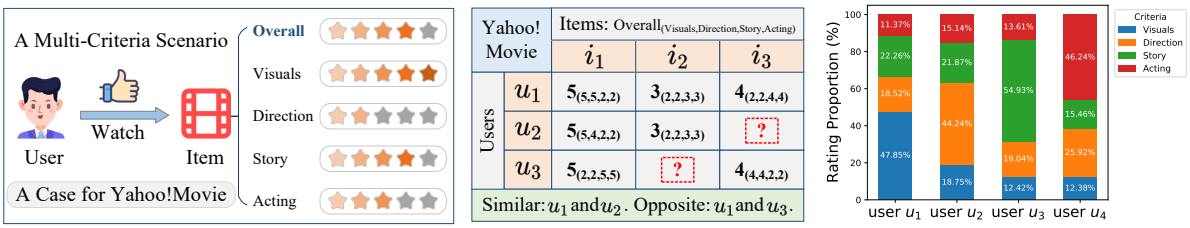
1 INTRODUCTION

Recommender systems (RSs) are critical technologies for alleviating information overload in the Internet era [5, 31]. Its essence is to provide users with personalized items (e.g., products, movies, news, etc.) by analyzing their historical behaviors and preferences. In the research of recommendation methods, collaborative filtering (CF) has become a prominent mainstream technique [67], which generates recommendation results by exploiting the similarity between users and items according to their interaction records [8]. In recent years, with the development of deep learning [34], there has been a shift in recommendation methods from matrix factorization to neural network-based modeling [57]. For example, cutting-edge approaches such as reinforcement learning

*The corresponding author.

Authors' Contact Information: Zhihao Guo, Shanxi University, Taiyuan, Shanxi, China, gzh081700@163.com; Peng Song, Shanxi University, Taiyuan, Shanxi, China, songpeng@sxu.edu.cn; Chenjiao Feng, Shanxi University of Finance and Economics, Taiyuan, Shanxi, China, fengcj@sxufe.edu.cn; Kaixuan Yao, Shanxi University, Taiyuan, Shanxi, China, ykx@sxu.edu.cn; Jiye Liang, Shanxi University, Taiyuan, Shanxi, China, ljy@sxu.edu.cn.

ACM 1558-2868/2025/8-ART
https://doi.org/10.1145/3757737



(a) A case about MC scenario. (b) MC ratings in Yahoo!Movie. (c) Selection bias.

Fig. 1. MC rating scenario and selection bias in Yahoo!Movie.

[3, 19], graph neural network [16, 68], and large language model [6, 71] have been introduced into RSs to facilitate high-quality decision making.

Although the existing methods exhibit outstanding performance, most of them are modeled based on a single user-item rating matrix (i.e., overall rating) for recommendation purposes [70]. Different from traditional single criterion methods, multi-criteria systems enhance the predictive quality of the model by introducing additional auxiliary information [1, 44]. For example, in the Yahoo!movie scenario (Fig. 1a), its rating system not only contains the overall rating, but also is subdivided into four criteria, including visuals, direction, story, and acting. Moreover, MC ratings reveal the heterogeneity of user preferences and can improve the accuracy of RSs [30, 41]. As an illustration, in Fig. 1b, although users u_1 and u_3 have similar overall rating, they exhibit opposite preferences in MC ratings. In contrast, u_1 and u_2 have more similar rating patterns. Therefore, RSs need to develop new technology to extract higher-order user preferences from MC ratings.

So far, the MC rating methods are still not fully explored. On the one hand, the existing studies usually introduce traditional similarity measures after splitting MC ratings and achieve prediction through an aggregation function (e.g., weighted summation) [33, 51]. This modeling scheme ignores the integrity of the criteria set and the heterogeneity of user preferences. On the other hand, users often tend to rate items they like or dislike. This spontaneous behavior may result in typical selection bias [11, 12], making the data collected not a representative sample set. In the case of Fig. 1c, we pick four representative users on the training set and normalize the proportion of ratings above the median for each user on different rating criteria. The results show significant differences in the rating criteria that users focus on. For example, user u_1 prefers the visuals, while u_3 pays more attention to the story. Traditional methods treat the training errors in all observation labels as a loss function to uncover the correlation between user feedback and model predictions [44]. This data-driven correlation learning paradigm may continuously amplify the selection bias of RSs, which may damage the model's recommendation quality and user experience [11, 65]. For example, the system will be more inclined to recommend items with high visuals ratings for user u_1, which ignores personalized MC preferences. Therefore, it is important to develop a recommendation framework that is applicable to MC ratings and captures the causality of user interactions [72, 78].

To remedy the limitations of the above methods, we describe MC scenarios based on causal inference [45] and leverage graph neural network (GNN) [22] to extract higher-order associations between users and items. Causal inference focuses on extracting causal relationships among variables from the target task, which can help RSs identify spurious association and mitigate the selection bias problem amplified [21, 27]. GNN is a neural network technique that excels in modeling non-Euclidean graph structure data [52, 75], which can help RSs learn complex MC behavioral characteristics of users [14, 69]. Therefore, we argue that integrating the strengths of causal inference and GNN may be an effective path to approach MC task and alleviating bias. We will address the following two critical challenges:

- **MC recommendation involves obvious bias problem.** The collected user behavior data is usually missing-not-at-random, which means that the observed samples are not a random subset of all possible user-item interaction pairs. For example, users tend to rate the items they like. This implies that the observed data cannot accurately reflect the underlying overall preference distribution of users. That is, the training data inherently contains bias. The traditional MC methods primarily estimate the rating probability conditioned on user and item representations. However, this training paradigm tends to inherit and continuously expand the bias problem as the model is iteratively updated in the feedback loop [39]. Therefore, how to design a new inference strategy with causal inference is the primary challenge for alleviating the bias problem.
- **MC rating data contain heterogeneity in terms of user behavioral preferences.** In the heterogeneous graph composed of MC scenes, each pair of nodes may contain different types of interactions among them. RSs should consider the complexity of MC behaviors when selecting items and develop comprehensive assessments according to the importance of different criteria preferences. In addition, the supervised signals suffer from severe data sparsity, which may bring difficulties to the training and limitation to the generalization ability of the model. Therefore, how to utilize GNN for multi-dimensional perspective information fusion and mining complex user preferences is another important challenge.

To address the above challenges, we propose a novel debiasing framework called Multi-Criteria Causal Recommendation (MCCR). Specifically, we adopt the structural causal model (SCM) to construct causal graph suitable for MC scenarios, and identify the back-door path opened by the confounder that causes bias amplified [45]. Subsequently, we implement interventions via the do calculus and utilize the back-door adjustment to design new training and inference paradigms for debiasing purpose. This unbiased estimation strategy not only captures the causal relationships between user behaviors and recommendation decisions, but also effectively alleviates the selection bias problem. Next, we construct a bipartite interaction graph corresponding to each criterion. Each graph defines the global embeddings of the nodes and the local embeddings in the single-criterion view, which are used to enhance the integration and information learning capabilities of the model. Based on this, we measure the degree of association among target rating and auxiliary criteria by using the graph attention mechanism [56], and extract the heterogeneity of user preferences by capturing their sensitivity to different criteria. Moreover, we design a self-supervised learning [40] loss in MC scenarios to enhance the embedding representation performance for overall rating interactions. Finally, losses co-optimization improves the robustness of the model and alleviates data sparsity in supervised signals. Our contributions are summarized below:

- We propose a causal graph for analyzing the causal relationships among the variables in MC recommendation and illustrate the real reason why selection bias is amplified. To the best of our knowledge, this is the first attempt to use causal inference to optimize MC methods.
- We develop an MCCR framework that efficiently encodes diversity information of each criterion to mine MC behavioral characteristics of users. This approach enhances the extraction of local and global structure in complex heterogeneous MC rating data.
- The experimental results on six public datasets show that the Top-N recommendation and debiasing performance of the proposed MCCR outperforms the existing baselines.

2 RELATED WORK

In this section, we review three representative categories of methods in RSs: CF-based recommendation methods [49], causal inference-based recommendation methods [72], and MC rating recommendation methods [2].

2.1 CF-based Recommendation Methods

CF methods [17, 73] realize personalized recommendation by analyzing the similar preferences of the user's historical behaviors, which are divided into two categories: matrix factorization-based methods [26], and neural network-based methods [25, 29]. In early research, CF mainly employs matrix factorization techniques, such as singular value decomposition, to learn the implicit features of users and items. For example, He et al. [28] designed a matrix factorization algorithm using alternating least squares to optimize implicit feedback data. With the development of neural network technology, CF methods leverage the advantages to fuse multi-source information (e.g., user and item attributes, clicks and comments, etc.) [18]. Currently, many deep learning methods have been applied to RSs to provide more accurate recommendation results. For example, Ahmadian et al. [4] proposed a reinforcement learning integration approach to formulate recommendation strategies based on prediction and credibility.

It is worth mentioning that GNN, as a technique for modeling complex topological relationships in graph-structured data, has been widely used in various recommendation scenarios [15, 58]. GNN can assist RSs in mining higher-order associations between users and items more deeply, and improve the prediction performance of the model. It designs corresponding graph network architectures (including homography, heterography, hypergraphs, dynamic graphs, and large-scale graphs) according to specific scenario types. For example, Qin et al. [46] proposed a graph ordinary differential equation framework to capture the underlying dynamics of user behavioral characteristics. Li et al. [35] proposed a multi-modal recommendation framework by leveraging knowledge distillation, which can capture the inherent bias among different modalities. However, the existing GNN-based methods model recommendation task by using the single-criterion mechanism, which ignores the MC behavioral characteristics of users in real life.

2.2 Causal Inference-based Recommendation Methods

Causal inference is remarkably effective in estimating causal effects among variables, and has been widely used in social science, medical research, artificial intelligence and other fields [47]. For RSs, causal inference can effectively alleviate the spurious association and bias problems (such as popularity bias [74], location bias [23], exposure bias [48], etc.) caused by confounding variables [11]. At present, recommendation methods based on causal inference are divided into two categories: the Rubin Causal Model (RCM)-based methods [47, 48], and the Structural Causal Model (SCM)-based methods [10, 20, 53, 63, 72].

The RCM describes the possible outcomes when individuals receive different treatments by using causal effect estimates. For example, Schnabel et al. [50] proposed an unbiased evaluator by using the inverse propensity score to correct for selection bias in the observed data. Song et al. [54] proposed a conservative doubly robust framework to mitigate the bias problem. The SCM employs causal graphs and structural equations to describe the process of generating data from a causal perspective. For example, Wang et al. [59] alleviated the bias effects of confounding by correcting unbalanced item distributions. Zhao et al. [74] proposed a time-aware debiasing framework and inferred sensitivity to popularity bias by intervening. Chen et al. [9] proposed a novel debiasing strategy to alleviate the bias problem caused by the traditional knowledge distillation paradigm. The SCM can also be generalized to other types of RSs. For example, in recommendation scenarios containing knowledge graphs, the SCM can alleviate the bias problem caused by structural information and similarity scores [66]. In out-of-distribution scenarios, the SCM can mitigate the impact of outdated interactions by intervening in user feature transfers [60].

The existing methods mainly model the bias problem with the single-criterion mechanism and pay less attention to MC data. Given the advantage that the SCM can describe the causal relationships among variables in detail from the data-generation perspective, we propose two improvements strategies. First, we argue that the causal relationships between user preferences and model decisions play a key role in alleviating the spurious association.

Therefore, we construct a causal graph to analyze the selection bias problem. Second, we design an unbiased estimation paradigm for the MC recommendation task, which improves the accuracy of the model.

2.3 MC Rating Recommendation Methods

The multi-dimensional user preferences included in MC ratings provide important decision support for RSs [30, 37]. Adomavicius et al. [2] summarized and defined a class of MC decision making problems in RSs. The existing MC rating recommendation methods are categorized into the heuristic-based methods [2] and the model-based methods [36].

In early MC research, the heuristic-based methods achieved prediction by using certain assumptions [43]. For example, some efforts [1, 13] measured MC preferences among users leveraging collaborative filtering, that is, similar users have similar preferences in the future. However, the sparse data characteristics greatly hinder the performance of these methods. In the subsequent development of MC recommendation, the model-based methods improved the performance of RSs by introducing some advanced techniques. For example, Li et al. [38] used multilinear singular value decomposition to explore the association between among criteria. Tallapally et al. [55] proposed an extended stacked self-encoder to efficiently model the relationship among user criteria and overall rating. Park et al. [44] made the first attempt to introduce GNN to MC scenarios to model collaborative signals through the constructed bipartite graph.

The existing MC methods mainly adopt the data-driven paradigm to learn the correlations in the data for decision-making purposes [1]. The opacity of this training paradigm may trigger spurious association that continually amplify the bias problem. Different from the above methods, we propose a debiasing framework through causal inference that leverages the back-door adjustment to construct new training and inference strategies for mitigating bias.

3 PROBLEM DESCRIPTION

In this section, we introduce the key notations used in this paper and illustrate the MC rating recommendation task through three definitions.

In general, let $\boldsymbol{\mathcal{U}} = \left\{u_1, u_2, \ldots, u_{|\boldsymbol{\mathcal{U}}|}\right\}$ and $\boldsymbol{\mathcal{I}} = \left\{i_1, i_2, \ldots, i_{|\boldsymbol{\mathcal{I}}|}\right\}$ denote the set of users and items, respectively, where $|\boldsymbol{\mathcal{U}}|$ and $|\boldsymbol{\mathcal{I}}|$ are the number of users and items, respectively. We describe the recommendation task in the MC rating scenarios by the following definitions:

Definition 1. *(MC Rating Matrices). The MC rating interaction record of users for items is represented as a set of matrices $\boldsymbol{\mathcal{R}} = \left\{\boldsymbol{\mathcal{R}}^0, \boldsymbol{\mathcal{R}}^1, \ldots, \boldsymbol{\mathcal{R}}^K\right\}$, where $\boldsymbol{\mathcal{R}}^0$ represents the overall rating matrix (target rating), $\left\{\boldsymbol{\mathcal{R}}^1, \boldsymbol{\mathcal{R}}^2, \ldots, \boldsymbol{\mathcal{R}}^K\right\}$ is the set of rating matrices on the other K criteria (auxiliary criteria), $\boldsymbol{\mathcal{R}}^k \in \mathbb{R}^{|\boldsymbol{\mathcal{U}}|\times|\boldsymbol{\mathcal{I}}|}$, and $r^k_{u,i}$ represents the rating value of an item i by a user u in the kth criterion matrix $\boldsymbol{\mathcal{R}}^k$, $k \in \{0, 1, \ldots, K\}$.*

Definition 2. *(MC Interaction Graphs). Constructing heterogeneous user-item interaction graphs in MC scenarios based on $\boldsymbol{\mathcal{R}}$. We split the heterogeneous information of each criterion into a bipartite graph set $\boldsymbol{\mathcal{G}} = \left\{\mathcal{G}^0, \mathcal{G}^1, \ldots, \mathcal{G}^K\right\}$, where $\mathcal{G}^k = \left\{\boldsymbol{\mathcal{V}}, \boldsymbol{\mathcal{E}}^k\right\}$, $\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{U}} \cup \boldsymbol{\mathcal{I}}$ is the set of nodes, and $\boldsymbol{\mathcal{E}}^k$ is the set of edges (in this paper,we set that if $r^k_{u,i}$ is greater than the median, then there exists an edge connecting between u and i in $\mathcal{G}^k$). Based on this, this set $\boldsymbol{\mathcal{G}}$ of bipartite graphs is represented using a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{|\boldsymbol{\mathcal{U}}|\times|\boldsymbol{\mathcal{I}}|\times|K+1|}$, where $x^k_{u,i} = 1$ if the edge represented by $r^k_{u,i}$ exists, and $x^k_{u,i} = 0$ otherwise.*

Definition 3. *(MC Top-N Recommendation Task). Given the sets of users $\boldsymbol{\mathcal{U}}$ and items $\boldsymbol{\mathcal{I}}$, and the MC tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{|\boldsymbol{\mathcal{U}}|\times|\boldsymbol{\mathcal{I}}|\times|K+1|}$, the goal of MC Top-N recommendation is to predict the interaction probability of items that do not interact with u, i.e., $\hat{r}_{u,i} = f(u, i)$, where $f(\cdot)$ is the prediction function of the recommendation model and $\hat{r}_{u,i}$ is the interaction probability of u with i. Finally, the set of top N ranked items is recommended for u based on all the predicted values $\hat{r}_{u,i}$.*

Table 1. Summary of Key Notations.

Notations	Description
u, $\boldsymbol{\mathcal{U}}$	The user u and the sample space of users;
i, $\boldsymbol{\mathcal{I}}$	The item i and the sample space of items;
K	The number of criteria;
$\boldsymbol{\mathcal{R}}$	The set of MC rating matrices;
$\boldsymbol{\mathcal{R}}^0$	The overall rating matrix;
$\boldsymbol{\mathcal{R}}^k$	The rating matrix of the kth criterion;
$\boldsymbol{\mathcal{G}}$	The set of user-item interaction graphs;
$\mathcal{G}^0$	The interaction graph corresponding to the overall rating matrix;
$\mathcal{G}^k$	The interaction graph corresponding to the rating matrix of the kth criterion;
$\boldsymbol{\mathcal{V}}$, $\boldsymbol{\mathcal{E}}^k$	The set of nodes and the set of edges in $\mathcal{G}^k$;
$r^k_{u,i}$	The rating of item i by user u in $\boldsymbol{\mathcal{R}}^k$;
$\mathbf{e}_u$, $\mathbf{e}^k_u$	The global embedding of u and the local embedding of the kth criterion view;
$\mathbf{e}_i$, $\mathbf{e}^k_i$	The global embedding of i and the local embedding of the kth criterion view;
$\boldsymbol{\alpha}$	The attention coefficient among the different criteria;
e_u, e_i	The final embedding of user u and the final embedding of item i;
m^k_u	The preference of user u for criterion k;
L	The number of GNN layers;
η	The BPR loss coefficient for the MC ratings;
λ_1	The loss coefficient for self-supervised learning;
τ	The temperature coefficient;

Table 1 lists the key notations used in this paper and their descriptions.

4 METHODOLOGY

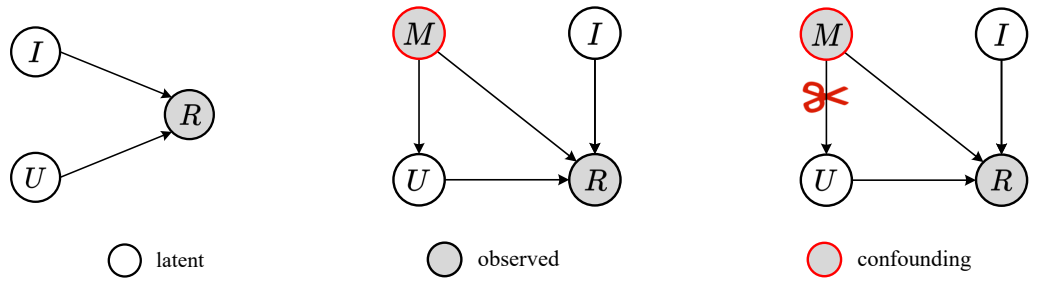
In this section, we first propose a causal perspective in MC recommendation scenarios to explain why selection bias is amplified and design a causal intervention strategy to mitigate the bias problem (see subsection 4.1). Next, we construct a modeling framework suitable for the MC rating tasks (see subsection 4.2). Finally, we introduce the optimization objective of the proposed MCCR (see subsection 4.3).

4.1 A Causal View of MC Scenarios

To achieve the purpose of mitigating bias in the MC recommendation task, we construct a causal graph based on the SCM from the perspective of data generation to explore the causal relationships between user feedback and model predictions.

4.1.1 Causal Graph. Fig. 2a illustrates the causal graph of the traditional methods, which achieves prediction based on the user-item matching mechanism. In this causal graph, U and I (latent variables) denote learned user representation and item representation, respectively, and R (observed variable) denotes the rating information, including MC ratings and overall rating. For example, many classical models [44] usually predict the rating R by calculating the inner product of the user representation U and the item representation I.

To investigate the bias problem in the MC scenarios, we modify the causal graph, as illustrated in Fig. 2b, and introduce the variable M (observed confounding) to represent the users' historical preference distribution over the K criteria. In the collected observation data, users exhibit significant differences in their rating behaviors for different criteria. Specifically, the frequency of user ratings on the criteria show an obvious imbalanced distribution (i.e., data bias). During model training, M directly affects the learning of user representations U, making it tend to reinforce the criterion of high-frequency ratings. This is attributed to the fact that U is optimized



(a) Traditional methods. (b) Consider confounding. (c) Cut the back-door path.

Fig. 2. Causal graphs for MC recommendation scenarios. The hollow circle indicates latent variables, the solid circle indicates observed variables, and the red circle indicates confounding variable. Specifically, U and I denote the user representation and the item representation to be learned (latent), respectively, R denotes the rating information (observed), including MC ratings and overall rating, and M denotes the users' historical preference distribution over the K criteria (confounding).

to fit the imbalanced historical data. As the model is trained iteratively, the bias problem will be gradually amplified in the feedback loop. For example, for K criteria, the criterion that is rated more times in historical behavior will obtain a higher prediction score.

Next, we reveal the reasons behind bias amplification from a causal perspective. It can be seen in Fig. 2b that two paths are formed from U to R: $U \rightarrow R$ and $U \leftarrow M \rightarrow R$. In general, path $U \rightarrow R$ is used to capture the loyalty preferences of users. However, path $U \leftarrow M \rightarrow R$ results in higher predictive scores for the high-frequency rating criterion, which significantly increases the likelihood of exposure for the corresponding items. According to the causal theory of the SCM [45], M opens the back-door path $U \leftarrow M \rightarrow R$ as a confounder, which may generate spurious association in the estimations between U and R. Therefore, avoiding the influence of exposure mechanism on the model when estimating user preferences is crucial to mitigate the bias problem. The explanations of all variables and edges are as follows:

- Node U denotes the user representation. For a user u, the representation $\mathbf{e}_u^k \in \mathbb{R}^d$ on the criterion view k is an ID-based embedding vector, where d is the embedding dimension.
- Node I denotes the item representation. For example, $\mathbf{e}_i^k \in \mathbb{R}^d$ is the representation of item i on the criterion view k.
- Node M denotes the users' historical preference distribution over the K criteria. Specifically, we formalize M by normalizing the frequency of ratings that exceed the median for each criterion in the training set, i.e., $\boldsymbol{M} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{|\mathcal{U}|})^T$, $\boldsymbol{M} \in \mathbb{R}^{|\mathcal{U}| \times K}$, where $\boldsymbol{m}_u \in \mathbb{R}^K$ represents the preference distribution of user u over K criteria. For example, in Fig. 1c, $K = 4$, and for user u_1, $\boldsymbol{m}_1 = [0.1137, 0.2226, 0.1852, 0.4785]$.
- Node R denotes the rating information, including the MC ratings and the overall rating.
- Edge $M \rightarrow U$ indicates that the users' historical preference distribution affects the learning of the user representation U, which is attributed to the fact that the model is optimized to fit imbalanced behavioral data. This optimization mechanism amplifies the bias problem in the feedback loop, leading to a shift of the learned representation towards the space dominated by the high-frequency rating criterion. For example, in Fig. 1c, user u_1 prefers the movie's visuals. With the iterative training of the model, RSs may transiently recommend visual movies to u_1.
- Edges $U \rightarrow R$, $I \rightarrow R$, and $M \rightarrow R$ represent that the user, the item, and the user's MC preferences jointly determine the final interaction probability. For example, a user u evaluates an item i from multiple dimensions based on their MC preferences, and obtains MC ratings and overall rating.

To eliminate the negative impact of spurious association on the model, we extend the causal graph to Fig. 2c by cutting off the back-door path $U \leftarrow M \rightarrow R$. Different from the correlation modeling paradigm of previous works [44, 51], this paper aims to identify the causal effect between U and R to achieve unbiased estimation in MC rating recommendation. Fortunately, the back-door adjustment provides a viable solution for this purpose. The causal theory [45] proves that the intervention probability after cutting off the back-door path can be estimated from the observed data. This means that we can infer unbiased interaction probabilities from historically collected MC data through the back-door adjustment without any actual intervention (see Section 4.1.3 for details).

4.1.2 Bias Analysis. To explore the reason why bias is amplified in MC recommendation scenarios, we analyze the modeling paradigm of the existing MC works according to Fig. 2b. Traditional methods employ data-driven modeling to estimate the interaction probability $P(R|U,I)$,

$$\begin{aligned} P\left(R|U,I\right) &\overset{(a)}{=} \sum_{m} P\left(R,m|U,I\right) \\ &\overset{(b)}{=} \sum_{m} P\left(R|U,I,m\right) P\left(m|U,I\right) \\ &\overset{(c)}{=} \sum_{m} P\left(R|U,I,m\right) P\left(m|U\right) \\ &\overset{(d)}{=} \sum_{m} P\left(R|U,I,m\right) P\left(U|m\right) P\left(m\right), \end{aligned} \tag{1}$$

where step (a) follows the law of total probability, i.e., summing over all possible values of M; Step (b) decomposes the joint probability $P\left(R,m|U,I\right)$ as the product of $P\left(R|U,I,m\right)$ and $P\left(m|U,I\right)$; In step (c), M and I are independent of each other according to Fig. 2b, therefore $P\left(m|U\right) = P\left(m|U,I\right)$; Step (d) follows the Bayes rule.

Due to the disturbance of $P\left(U|m\right)$, the probability $P\left(R|U,I\right)$ will be dominated by the user's historical preferences. That is, users are more inclined to select items that match with their historical interests, and these items will have a higher probability of being exposed. In this scenario, $P\left(U|m\right)$ causes spurious association, and the bias problem of RSs will become more and more serious after continuous iterative training. Therefore, it is key to alleviate bias by changing the exposure strategy during the inference stage, which will allow each item to be recommended fairly.

4.1.3 Back-Door Adjustment Strategy. To achieve debiasing by identifying the causal effect between U and R, we estimate the impact of the intervention $do\left(U,I\right)$ on R with the back-door adjustment (Fig. 2c). $P\left(R|do\left(U,I\right)\right)$ is the conditional probability after blocking the back-door path $U \leftarrow M \rightarrow R$, and the adjustment formula is derived as

$$\begin{aligned} P\left(R|do\left(U,I\right)\right) &\overset{(a)}{=} P_d\left(R|U,I\right) \\ &\overset{(b)}{=} \sum_{m} P_d\left(R|U,I,m\right) P_d\left(m|U,I\right) \\ &\overset{(c)}{=} \sum_{m} P_d\left(R|U,I,m\right) P_d\left(m\right) \\ &\overset{(d)}{=} \sum_{m} P\left(R|U,I,m\right) P\left(m\right), \end{aligned} \tag{2}$$

where step (a) is the estimation of the manipulation probability $P_d\left(\cdot\right)$ after cutting off the $U \leftarrow M \rightarrow R$ by $do\left(U,I\right)$; step (b) follows Bayesian theory; step (c) follows Fig. 2c, where U and I are independent of M, i.e., $P_d\left(m\right) = P_d\left(m|U,I\right)$; in step (d), the marginal probability $P\left(m\right)$ is invariant before and after the intervention,

i.e., $P(m) = P_d(m)$, and the conditional probability $P(R|U, I, m)$ is invariant due to the fact that the response function of R with U, I, and M is fixed whether U changes spontaneously or is manipulated to change by the intervention, i.e., $P(R|U, I, m) = P_d(R|U, I, m)$.

Inspired by [72], we estimate $P(R|do(U, I))$ from both training and inference stages:

Train. In the training phase, we predict the corresponding interaction probability based on the rating information of each criterion. For criterion k, the probability $P(R|U, I, m)$ is estimated given $U = \mathbf{e}_u^k$, $I = \mathbf{e}_i^k$, and the user's preference m_u^k for the criterion,

$$\begin{aligned} & P\left(R = r_{u,i}^k | U = \mathbf{e}_u^k, I = \mathbf{e}_i^k, m = m_u^k, \right) \\ & \overset{(a)}{=} f\left(\mathbf{e}_u^k, \mathbf{e}_i^k, m_u^k\right) \\ & \overset{(b)}{=} \text{LeakyReLU}\left((\mathbf{e}_u^k)^T \mathbf{e}_i^k\right) \times \text{Sigmoid}\left((m_u^k)^\gamma\right), \end{aligned} \tag{3}$$

where $f(\cdot)$ in step (a) is the learning framework of the model, and we employ a decoupling manner as shown in step (b). This decoupling is applicable to any recommendation model backbone, which ensures the generality of the proposed debiasing framework. $\mathbf{e}_u^k$ and $\mathbf{e}_i^k$ are representations of user u and item i in criterion k, respectively. m_u^k denotes the degree of user preference for criterion k. LeakyReLU $(\cdot)$ and Sigmoid $(\cdot)$ are the activation functions, and γ is a hyperparameter.

We approximate the user's historical preference distribution m_u^k on the criterion k as

$$\begin{aligned} m_u^k &= \frac{exp\left(\text{LeakyReLU}(q_u^k)\right)}{\sum_{t=1}^K exp\left(\text{LeakyReLU}(q_u^t)\right)}, \\ q_u^k &= N_u^k / N_u^0, \\ N_u^k &= \textstyle\sum_{i \in \mathcal{N}_u^k} \mathbb{I}(r_{u,i}^k > \text{median}^k), \\ N_u^0 &= \textstyle\sum_{i \in \mathcal{N}_u^0} \mathbb{I}(r_{u,i}^0 > \text{median}^0), \end{aligned} \tag{4}$$

where q_u^k denotes the rating frequency of user u on criterion k, N_u^k and N_u^0 denote the number of interactions of the training set on criterion view k and overall rating view, respectively, $\mathcal{N}_u^k$ denotes the item set that u has interacted with on view k, median^k denotes the median of the rating ranges on view k, $\mathbb{I}(\cdot)$ is an indicator function, and $\mathbb{I}(\cdot) = 1$ if $(\cdot)$ is true, otherwise, $\mathbb{I}(\cdot) = 0$.

Inference. In the inference phase, the goal of this paper is to achieve unbiased estimation according to the intervention probability $P(R|do(U, I))$. Equation 1 indicates that the traditional correlation modeling paradigm $P(R|U, I)$ may result in bias being continuously amplified during the model iteration process. Therefore, we hope to correct the bias problem of the model by changing the exposure mechanism of the items. Formally,

$$\begin{aligned} & P\left(R = r_{u,i}^k | do\left(U = \mathbf{e}_u^k, I = \mathbf{e}_i^k\right)\right) \\ & \overset{(a)}{=} \sum_{m_u^k} P\left(r_{ui}^k | \mathbf{e}_u^k, \mathbf{e}_i^k, m_u^k\right) P\left(m_u^k\right) \\ & \overset{(b)}{\approx} f\left(\mathbf{e}_u^k, \mathbf{e}_i^k, \textstyle\sum_{m_u^k} (m_u^k)^\gamma P(m_u^k)\right) \\ & \overset{(c)}{=} \text{LeakyReLU}\left((\mathbf{e}_u^k)^T \mathbf{e}_i^k\right) \times \text{Sigmoid}\left(\mathbb{E}(M)^\gamma\right), \end{aligned} \tag{5}$$

where $\mathbb{E}(M)$ is the expectation of the user's historical preference M. The inference strategy allows each item to be fairly exposed without being interfered by $P(U|m)$. Different from the traditional single-criterion methods, it

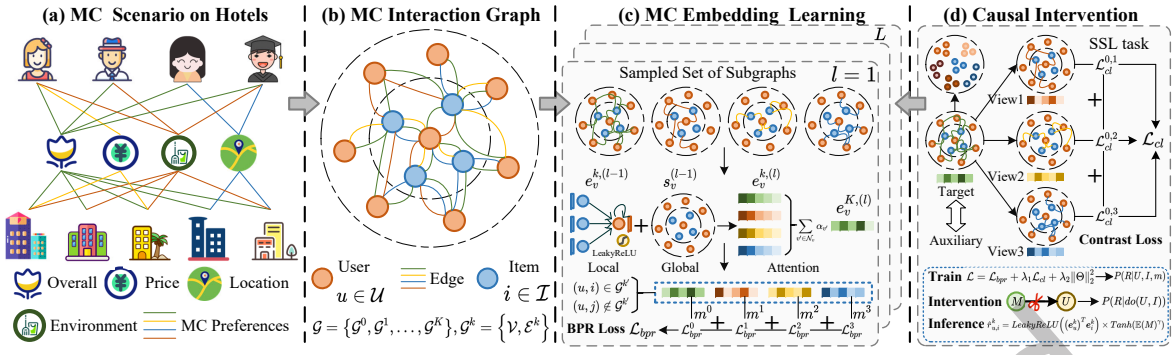


Fig. 3. The overall framework of the MCCR. (a) In a MC case on hotel recommendations, four rating criteria are included. (b) We construct the interaction graph set $\mathcal{G}$ based on the overall rating and each criterion. (c) In $\mathcal{G}^k$, the local embedding $\mathbf{e}_v^{k,(l)}$ and global embedding $\mathbf{e}_v^{(l)}$ of node neighbors are aggregated to generate the vector $\mathbf{e}_v^k$ in a specific view, and the dependency among the criteria is captured based on the attention mechanism to get the representation $\mathbf{e}_v$ in the target view, and the BPR loss $\mathcal{L}_{bpr}$ is constructed. (d) The contrastive loss $\mathcal{L}_{cl}$ among the target view and the auxiliary criteria is constructed. Finally, we joint $\mathcal{L}_{bpr}$ and $\mathcal{L}_{cl}$ and achieve optimization and prediction according to the training and inference strategies Eqs. 3 and 5.

is necessary to develop a framework that can specifically deal with MC ratings for the recommendation backbone model $f(\cdot)$. In section 4.2, we describe how to implement $f(\cdot)$ to build a recommendation framework for MC scenarios.

4.2 MC-aware Graph Representation Learning

In this subsection, we develop the MCCR framework for MC Top-N recommendation task, which exploits higher-order connectivity to recursively propagate the embedding representations. The overall architecture of the MCCR is shown in Fig. 3. The MCCR learns the embedding representations of users and items based on GNN, and its framework contains two modules, single-criterion feature aggregation module and MC information propagation module, respectively.

4.2.1 Single-Criterion Feature Aggregation Module. This module is used to model the representations of users and items on each criterion view. Specifically, we adopt graph convolution operations to aggregate feature information from the neighborhood of node and update the embeddings in the criterion-specific interaction graph. For convenience, we employ v to denote a node in the view (either a user or an item). In view $\mathcal{G}^k$, the embedding of v in the lth layer is obtained by the aggregation function $\mathbf{e}_v^{k,(l)} = g(\mathbf{e}_v^{(l-1)}, \mathbf{e}_{\mathcal{N}_v^k}^{k,(l-1)})$,

$$\mathbf{e}_v^{k,(l)} = \sigma(\mathbf{e}_v^{(l-1)} + \mathbf{W}_{Mean}^{(l)} Mean(\left\{\mathbf{e}_{v'}^{k,(l-1)}, \forall v' \in \mathcal{N}_v^k\right\})), \tag{6}$$

where $v \in \{u, i\}$, $\mathbf{e}_v^{k,(l)}$ is the embedding of v in the lth layer in $\mathcal{G}^k$, $\mathbf{e}_v^{(l-1)}$ is the shared global embedding for information transfer among criteria, $\mathbf{e}_{v'}^{k,(l-1)}$ is the local embedding for capturing user preferences in a specific criterion, $\mathbf{W}_{Mean}^{(l)} \in \mathbb{R}^{d \times d}$ is the weight matrix, $\mathcal{N}_v^k$ is the set of neighbors of v, and $Mean(\cdot)$ computes the average of all neighbor embeddings, $\sigma(\cdot)$ is the ReLU activation function.

4.2.2 MC Information Propagation Module. This module models the heterogeneity of user preferences in higher-order MC ratings by aggregating the embedding representations under different criterion views. After coding the embeddings of $\mathcal{G}^k$ for a particular criterion, we model the dependencies among the different criteria. For

Algorithm 1 The algorithm of the MCCR

Input: The MC graph set $\mathcal{G} = \{\mathcal{G}^0, \mathcal{G}^1, \ldots, \mathcal{G}^K\}$, where $\mathcal{G}^k = \{\mathcal{V}, \mathcal{E}^k\}$; The user's historical rating distribution $\boldsymbol{M} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_{|\mathcal{U}|})^T$, where $\boldsymbol{m}_u \in \mathbb{R}^K$
Output: Predicted probability $\hat{r}_{u,i} = f(u, i)$
Initialize the embeddings $\mathbf{e}_v^{k,(l)}$ and $\mathbf{e}_v^{(l)}$;
while *MCCR not converge* **do**
  for $v = 1$ **to** $\mathcal{V} = |\mathcal{U} \cup \mathcal{I}|$ **do**
    Update $\mathbf{e}_v^{k,(l)}$ by Eq. 6 ;
    Integrate $\boldsymbol{C}_v^{(l)} \leftarrow \left\{\mathbf{e}_v^{1,(l)}, \mathbf{e}_v^{2,(l)}, \ldots, \mathbf{e}_v^{K,(l)}\right\}$;
    Calculate the attention $\boldsymbol{\alpha}_v^{(l)}$ by Eq. 8 ;
    Obtain $\mathbf{e}_v^{(l)}$ by Eq. 7 ;
    Obtain $\mathbf{e}_u, \mathbf{e}_i \leftarrow \mathbf{e}_u^{(l)}, \mathbf{e}_i^{(l)}$
  end
  Predict $\hat{r}_{u,i}^k \leftarrow \mathbf{e}_u^k, \mathbf{e}_i^k, m_u^k$;
  Calculate the BPR loss $\mathcal{L}_{bpr}$ by Eq. 12;
  Calculate the CL loss $\mathcal{L}_{cl}$ by Eq. 13;
  Obtain loss $\mathcal{L} \leftarrow \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cl} + \lambda_2 \|\Theta\|_2^2$;
  Update parameters with Adam
end
Inference $\hat{r}_{u,i} \leftarrow \mathbf{e}_u, \mathbf{e}_i, \mathbb{E}(M)$

any node v, $\boldsymbol{C}_v^{(l)} = \left\{\mathbf{e}_v^{1,(l)}, \mathbf{e}_v^{2,(l)}, \ldots, \mathbf{e}_v^{K,(l)}\right\}$ denotes its set of embeddings under K criteria, where $\boldsymbol{C}_v^{(l)} \in \mathbb{R}^{K \times d}$. Considering the differences in users' preferences, we introduce a graph attention mechanism to measure the degree of association among the target rating and the auxiliary criteria,

$$\mathbf{e}_v^{(l)} = \sigma(\mathbf{e}_v^{0,(l-1)} + ((\boldsymbol{\alpha}_v^{(l-1)})^T \boldsymbol{C}_v^{(l-1)})^T), \tag{7}$$

where $\boldsymbol{\alpha}_v^{(l-1)} \in \mathbb{R}^{K \times 1}$ is the matrix of attention coefficients, which reflects the dependence of v with respect to the other criteria. $\boldsymbol{\alpha}_v^{(l-1)}$ is calculated as:

$$\boldsymbol{\alpha}_v^{(l)} = \text{Softmax}(\text{Sigmoid}((\mathbf{W}_1^{(l)}(\boldsymbol{C}_v^{(l)})^T)^T \mathbf{W}_2^{(l)} \mathbf{e}_v^{0,(l)})), \tag{8}$$

where Softmax $(\cdot)$ is used for normalization , $\mathbf{W}_1^{(l)} \in \mathbb{R}^{d \times d}$ and $\mathbf{W}_2^{(l)} \in \mathbb{R}^{d \times d}$ are the parameter matrices to be learned.

To inject different layers of higher-order features into the node's embedding learning, we leverage mean pooling to drive a context-aware propagation structure that obtains the final embeddings of users and items,

$$\mathbf{e}_u = \frac{1}{L}\sum_{l=0}^{L} \mathbf{e}_u^{(l)}, \quad \mathbf{e}_i = \frac{1}{L}\sum_{l=0}^{L} \mathbf{e}_i^{(l)}, \tag{9}$$

where $\mathbf{e}_u$ and $\mathbf{e}_i$ denote the embedding of user u and the embedding of item i after aggregating MC feature information, respectively.

4.3 Prediction and Optimization

In this subsection, we introduce the prediction and optimization objectives of the MCCR. For criterion k, we predict the interaction probability between u and i according to Eq. 3 derived from the framework $f(\cdot)$,

$$\hat{r}_{u,i}^{k} = \text{LeakyReLU}\left((\mathbf{e}_u^k)^T \mathbf{e}_i^k\right) \times \text{Sigmoid}\left((m_u^k)^\gamma\right), \tag{10}$$

where $\hat{r}_{u,i}^{k}$ is the predicted value, which represents the potential preference of the user u for the item i on criterion k.

We construct the Bayesian Personalized Ranking (BPR) loss in $\mathcal{G}^k$,

$$\mathcal{L}_{bpr}^{k} = -\sum_{(u,i,j)\in O^k} log\left(\sigma\left(\hat{r}_{u,i}^{k} - \hat{r}_{u,j}^{k}\right)\right), \tag{11}$$

where $O^k = \left\{(u,i,j) \mid (u,i) \in \mathcal{G}^k, (u,j) \notin \mathcal{G}^k\right\}$, $k \in \{0,1,\ldots,K\}$, and (u,j) is a randomly sampled set of negative sample pair.

The overall BPR loss is

$$\mathcal{L}_{bpr} = \mathcal{L}_{bpr}^{0} + \eta \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{bpr}^{k}, \tag{12}$$

where η is a hyperparameter, which is used to adjust the strength of the MC ratings for model update during the training process.

In addition, we design a self-supervised contrastive loss among the overall rating and the criteria. This loss makes it possible for the MCCR to obtain more robust representations of users and items by maximizing the consistency among the different criteria. Thus, even in data sparse scenarios, the MCCR can utilize knowledge transfer among criteria to make effective and accurate recommendations. The self-supervised loss on criterion k is

$$\mathcal{L}_{cl}^{k} = -\sum_{v\in\mathcal{V}} log \frac{exp\left(s(\mathbf{e}_v, \mathbf{e}_v^k)/\tau\right)}{\sum_{v'\notin\mathcal{V}, v\neq v'} exp\left(s(\mathbf{e}_v, \mathbf{e}_{v'}^k)/\tau\right)}, \tag{13}$$

where τ is a hyperparameter for controlling contrast intensity, and $s(\cdot)$ is a cosine similarity function to compute positive or negative samples. The overall contrastive loss is $\mathcal{L}_{cl} = \mathcal{L}_{cl}^{1} + \mathcal{L}_{cl}^{2} + \cdots + \mathcal{L}_{cl}^{K}$.

Finally, the optimization objective for training is obtained as:

$$\mathcal{L} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cl} + \lambda_2 \left\|\Theta\right\|_2^2, \tag{14}$$

where λ_1 is the hyperparameter and λ_2 is used for regularization to prevent overfitting. For the inference of the model, we realize it by $\hat{r}_{u,i} = \text{LeakyReLU}\left((\mathbf{e}_u)^T \mathbf{e}_i\right)$ and perform Top-N unbiased recommendations.

The pseudo-code of the MCCR is shown in Algorithm 1.

5 EXPERIMENTAL RESULTS AND ANALYSIS

We validate the performance of MCCR by exploring the following four research questions:

- **RQ1**: How does the MCCR perform in MC scenarios compared to the existing data-driven methods?
- **RQ2**: How does the debiasing performance of the MCCR compare to the existing methods based on causal inference?
- **RQ3**: How do different components affect the recommendation performance of the MCCR?
- **RQ4**: How do differences in hyperparameter settings affect the MCCR performance?

Table 2. Dataset statistics. #Overall and #MC denote the interaction numbers of overall rating and MC ratings, respectively, and K denotes the number of criteria.

Statistics	TripAdvisor	Yahoo!Movie	RateBeer	Yelp-2022
#Users	4,265	1,821	4,017	58,971
#Items	6,275	1,472	3,422	19,820
#Overall	34,383	46,176	159,755	445,724
#MC	202,859	175,468	607,067	1,408,487
K	7	4	4	3
Sparsity	1.27E-03	2.07E-02	1.39E-02	5.29E-04

5.1 Experimental Settings

5.1.1 Datasets. We conduct our experiments on four MC datasets. These datasets include rating information of different criteria:

- **TripAdvisor**[1] statistics from the travel website contain rating data for hotels around the world. This includes overall rating and ratings for seven criteria such as *business*, *quality*, *cleanliness*, *location*, *rooms*, *service* and *value*. All ratings range from 1 to 5 on the scale.
- **Yahoo!Movie**[2] comes from the online movie platform and includes an overall rating and four criteria: *story*, *acting*, *direction* and *visuals*. All rating criteria range from 1 to 5.
- **RateBeer**[3] is about beer ratings and contains an overall rating and ratings by four criteria: *appearance*, *aroma*, *taste*, and *palate*. The ratings vary from 1 to 5 (*appearance* and *palate*), 1 to 10 (*aroma* and *taste*), and 1 to 20 (overall rating).
- **Yelp-2022**[4] provides rating information about restaurants and includes interactive information on several criteria, such as the number of votes for the criteria *cool*, *funny*, and *useful*, in addition to an overall rating on a scale of 1 to 5.

We construct unbiased test environment by adopting the classical random splitting strategy. That is, each item in the test set has an equal probability of being selected. We convert the data format to implicit feedback in constructing the graph-structured data. When an interaction is marked as 1, it means that the user has positively evaluated the item. There are $K + 1$ graphs constructed, including one graph corresponding to the overall rating and K graphs corresponding to the MC ratings. The positive rating threshold for each dataset is set as their median. We randomly select a negative sample labeled 0 for training. Table 2 shows the statistics of the four datasets.

5.1.2 Baselines. We compare the MCCR with 16 baselines, including the data-driven recommendation methods (both the single-criterion recommendation methods and the MC recommendation methods) and the causal inference-based recommendation methods. For **the single-criterion methods**, we select five representative GNN-based models and train them only on the overall rating matrix. For **the MC recommendation methods**, we select six state-of-the-art models proposed in recent years. For **the causal inference-based recommendation methods**, we chose two classical RCM methods and three popular SCM methods.

The single-criterion recommendation methods

- **GC-MC** [7] employs graph autoencoder for link prediction to achieve recommendation.
- **SpectralCF** [76] models collaborative filtering tasks by spectral convolution on a graph.
- **NGCF** [61] utilizes graph neural networks to propagate collaborative signals in embedding learning.
- **DGCF** [62] model fine-grained user intents by disentangling graph collaborative filtering.

[1]http://tripadvisor.com/
[2]http://movies.yahoo.com/
[3]https://www.ratebeer.com
[4]https://www.yelp.com/dataset

- **LightGCN** [24] is an efficient graph convolutional recommendation framework.

The MC recommendation methods.

- **UBM** [77] uses a ranking strategy to model the MC recommendation task.
- **DMCF** [42] implements the MC collaborative filtering model with deep neural networks.
- **AEMC** [51] mines user preferences in MC scenarios with deep autoencoder.
- **CFM** [13] predicts overall user ratings by automatically weighting MC ratings.
- **LightGCN-MC** [24] is LightGCN being applied to MC rating scenarios.
- **CPA-LGC** [44] mines users' MC preferences and complex higher-order relationships with graph convolutional neural networks.

The causal inference-based recommendation methods.

- **IPW** [50] is an inverse probability weighting method for dealing with sample imbalance or selection bias.
- **DR** [32] is a double robust method that combines IPW and regression models.
- **PDA** [72] is a causal method used to alleviate the popularity bias.
- **DecRS** [59] is a backdoor adjustment method used to mitigate amplified bias.
- **DCF** [64] is a deconfounding collaborative filtering method based on multiple causal inference.

5.1.3 Evaluation Metrics. To validate the Top-N performance of the MCCR, we use three common evaluation metrics in RSs: Hit Ratio (HR), Recall, and Normalized Discounted Cumulative Gain (NDCG). The three metrics measure the accuracy of the model and the ranking quality of Top-N recommendations. HR is used to evaluate whether the model successfully predicts the items interacted with in the user's real list. Formally,

$$\text{HR@N} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \phi(R(u) \cap T(u) \neq \emptyset), \tag{15}$$

where $T(u)$ represents the real interaction list, $R(u)$ represents the recommendation list, $\emptyset$ represents the empty set, and $\phi(\cdot)$ represents an indicator function that $\phi(\cdot) = 1$ if $\cdot$ is true, otherwise $\phi(\cdot) = 0$.

Recall is used to measure the proportion of all positive samples that are correctly recommended by the model. Formally,

$$\text{Recall@N} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{|R(u) \cap T(u)|}{|T(u)|}. \tag{16}$$

NDCG is used to evaluate the ranking order and true relevance of recommended items, not just hits. Formally,

$$\text{NDCG@N} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \left(\frac{1}{\sum_{i=1}^{\min(|T(u)|, N)} \frac{1}{\log_2(i+1)}} \sum_{i=1}^{N} \frac{\phi(R(u)_i \in T(u))}{\log_2(i+1)} \right). \tag{17}$$

5.1.4 Hyperparameter Settings. For a fair comparison, we set the experimental parameters uniformly. The datasets are divided into 80% training set and 20% test set. All models employ the Adam optimizer to train the network. The mini-batch size is set as 2048. The embedding dimension is set as 64. The learning rate is searched in the range $\{1e-4, 1e-3, 1e-2\}$. The number of layers L is tuned in the range $\{2, 4, 6, 8\}$. The L_2 regularization coefficient λ_2 is tuned in the range $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$. In the MCCR, the BPR loss coefficient η and contrast loss coefficient λ_1 are tuned in the range $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, and the temperature coefficient τ is tuned in the range $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$. The Top-N recommendation list is set as $N = 20$ and $N = 50$. For specific hyperparameters in the baseline models, we follow the recommendations of the original paper settings.

Table 3. Performance comparison of MCCR and data-driven models on TripAdvisor and Yahoo!Movie.

Datasets	TripAdvisor						Yahoo!Movie					
	Top-20			Top-50			Top-20			Top-50		
Model	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
GC-MC [7]	0.0746	0.0368	0.0135	0.1118	0.0604	0.0238	0.4665	0.1512	0.1603	0.6076	0.2668	0.1951
SpectralCF [76]	0.0652	0.0322	0.0396	0.1025	0.0561	0.0554	0.4597	0.1496	0.1637	0.5913	0.2619	0.1983
NGCF [61]	0.1073	0.0491	0.0411	0.1414	0.0702	0.0552	0.4532	0.1443	0.1846	0.5825	0.2583	0.2092
DGCF [62]	0.1104	0.0504	0.0621	0.1537	0.0776	0.0761	0.5173	0.1586	0.1866	0.7002	0.2713	0.2083
LightGCN [24]	0.1239	0.0626	0.0633	0.1709	0.0845	0.0767	0.5415	0.1626	0.1819	0.7386	0.2807	0.2097
UBM [77]	0.0922	0.0475	0.0382	0.1372	0.0697	0.0443	0.1519	0.0634	0.0727	0.3744	0.1453	0.1139
DMCF [42]	0.0756	0.0393	0.0145	0.1066	0.0595	0.0335	0.3038	0.0996	0.1252	0.4185	0.1788	0.1805
AEMC [51]	0.0734	0.0387	0.0236	0.1197	0.0622	0.0322	0.3157	0.1088	0.1244	0.4538	0.1957	0.1886
CFM [13]	0.1085	0.0519	0.0508	0.1433	0.0718	0.0569	0.4272	0.1375	0.1483	0.5235	0.2506	0.1968
LightGCN-MC [24]	0.1372	0.0662	0.0667	0.1868	0.0991	0.0786	0.5628	0.1759	0.1836	0.7552	0.2906	0.2175
CPA-LGC [44]	0.1441	0.0719	0.0843	0.1975	0.1032	0.0963	0.5651	0.1843	0.1927	0.7583	0.2955	0.2388
MCCR-GNN	0.1507	0.0764	0.0855	0.2021	0.1064	0.0981	0.5833	0.1918	0.2346	0.7669	0.3017	0.2657
MCCR(Ours)	**0.1662***	**0.0837***	**0.0904***	**0.2166***	**0.1108***	**0.1025***	**0.6419***	**0.2066***	**0.2853***	**0.7981***	**0.3234***	**0.3593***
%improv.	15.34%	16.41%	7.24%	9.67%	7.36%	6.44%	13.59%	12.10%	48.05%	5.25%	9.44%	50.46%

The bold score denotes the best experimental result and the underlined score indicates the best baseline. %improv. denotes the relative improvement of MCCR compared to the best baseline. "*" denotes statistically significant improvement compared to the best baseline (p-value < 0.01).

Table 4. Performance comparison of MCCR and data-driven models on RateBeer and Yelp-2022.

Datasets	RateBeer						Yelp-2022					
	Top-20			Top-50			Top-20			Top-50		
Model	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
GC-MC [7]	0.7592	0.3188	0.3005	0.8283	0.4523	0.3556	0.2477	0.1018	0.0795	0.3711	0.1225	0.0796
SpectralCF [76]	0.7465	0.3033	0.3022	0.8122	0.4461	0.3534	0.1366	0.0705	0.0532	0.2667	0.0897	0.0603
NGCF [61]	0.7551	0.3128	0.3065	0.8305	0.4597	0.3369	0.2879	0.1273	0.0857	0.3928	0.1427	0.0862
DGCF [62]	0.7325	0.2993	0.2995	0.7917	0.4332	0.3183	0.2886	0.1292	0.0809	0.3993	0.1486	0.0785
LightGCN [24]	0.7573	0.3165	0.3088	0.8129	0.4468	0.3505	0.2925	0.1337	0.0966	0.4009	0.1858	0.0967
UBM [77]	0.3925	0.1169	0.1539	0.5533	0.3256	0.1562	0.1539	0.0935	0.0457	0.3051	0.0922	0.0689
DMCF [42]	0.4733	0.1617	0.2116	0.6518	0.3705	0.2338	0.1416	0.0882	0.0556	0.2238	0.0869	0.0721
AEMC [51]	0.5886	0.2196	0.2879	0.7173	0.4053	0.2781	0.1354	0.0696	0.0671	0.2126	0.0835	0.0809
CFM [13]	0.6879	0.2768	0.2925	0.7664	0.4287	0.2993	0.2566	0.1072	0.0768	0.3897	0.1276	0.0828
LightGCN-MC [24]	0.7631	0.3291	0.3153	0.8671	0.4652	0.3557	0.2973	0.1359	0.0998	0.4115	0.2061	0.1034
CPA-LGC [44]	0.7866	0.3303	0.3225	0.8867	0.5044	0.3688	0.2985	0.1383	0.1019	0.4265	0.2297	0.1215
MCCR-GNN	0.8012	0.3397	0.3661	0.8952	0.5128	0.4626	0.3004	0.1397	0.1123	0.4291	0.2335	0.1383
MCCR(Ours)	**0.8557***	**0.3562***	**0.4166***	**0.9388***	**0.5309***	**0.5175***	**0.3152***	**0.1475***	**0.1275***	**0.4433***	**0.2456***	**0.1687***
%improv.	8.78%	7.84%	29.18%	5.88%	5.25%	40.32%	5.59%	6.65%	25.12%	3.94%	6.92%	38.85%

The bold score denotes the best experimental result and the underlined score indicates the best baseline. %improv. denotes the relative improvement of MCCR compared to the best baseline. "*" denotes statistically significant improvement compared to the best baseline (p-value < 0.01).

5.2 Overall Performance (RQ1)

In this subsection, we evaluate the performance of MCCR with the traditional data-driven methods and compare the performance of MCCR with the baselines in data-sparse scenarios.

5.2.1 Performance Comparison with the Data-driven Methods. We report the recommendation performance of the MCCR with the data-driven baselines on Top-20 and Top-50. Tables 3 and 4 show the overall performance of all models on the four MC datasets. We perform a t-test on the best baseline (p-value < 0.01) to ensure that the performance improvement of the MCCR is statistically significant. In addition, we report a variant of the MCCR, named **MCCR-GNN**, which removes the debiased inference strategy and implements recommendations based only on the proposed GNN framework. The **MCCR-GNN** is used to evaluate the effectiveness of the developed GNN framework in MC scenarios. We summarize the following conclusions according to the experimental results:

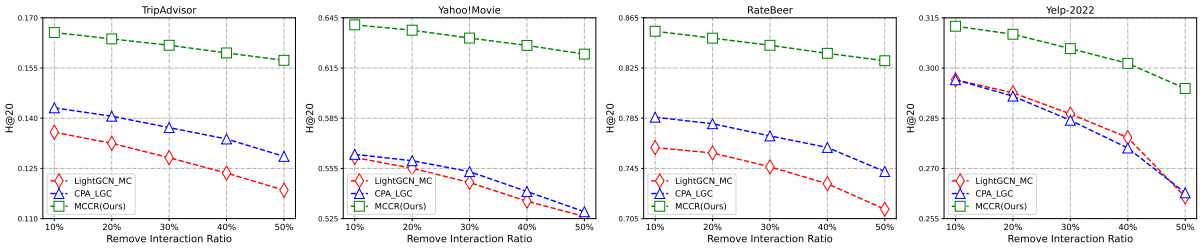
- In the four MC recommendation scenarios, the proposed MCCR consistently outperforms all baselines on three metrics. This improvement validates the effectiveness of the MCCR, which is attributed to its ability to mine complex user preferences and capture the causal relationships between user behavior and recommendation decisions. Compared to the best baseline, the MCCR achieves an average of 16.07% improvement across all datasets. Especially on Yahoo!Movie, the MCCR improves the metric N@50

by 50.46%. In addition, the superior performance of the MCCR validates our rationality in extracting higher-order heterogeneous relationships in MC data and mitigating bias.

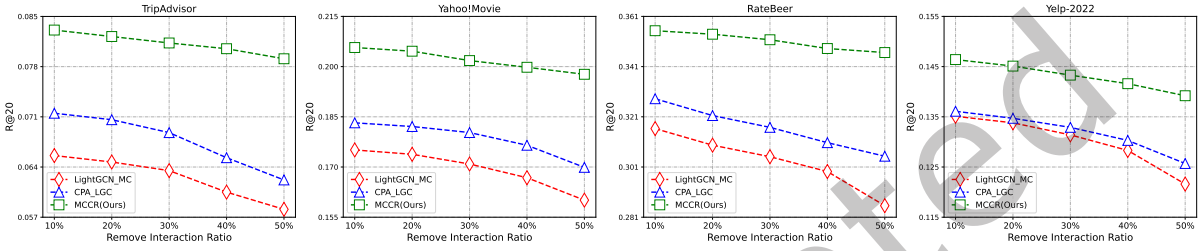
- Compared to a single overall rating, MC ratings can help the model generate better recommendation decisions. From the experimental results, we can see that LightGCN-MC consistently outperforms LightGCN in Top-N recommendation performance. This enhancement stems from fine-grained modeling of the heterogeneous user preferences contained in MC ratings. As described in the introduction, although two users have similar overall ratings, their preferences in MC ratings may be opposite. As a complement to the supervisory signals, the auxiliary information underlying MC ratings can effectively improve the recommendation quality and accuracy of the model during the modeling process. In Subsection 5.4, we verify the rationality of MC ratings in improving recommendation performance through more detailed experiments.
- Among the various MC recommendation methods, UBM, DMCF and AEMC perform poorly and even lag behind certain single-criterion methods (e.g. NGCF, DGCF, LightGCN, etc.). We believe that this may be due to the fact that the former model user preferences based on the learning paradigm of multi-layer perceptron, which is difficult to effectively capture the multi-dimensional behavioral characteristics of users. On the contrary, although the latter only make recommendation decisions based on the overall rating matrix, they utilize the GNN to model the user-item bipartite graph, which is able to effectively extract higher-order information from the user's historical interactions. These results indirectly reflect the advantages of GNN in modeling MC ratings.
- MCCR-GNN outperforms all baselines on the Top-N task. Even after removing the debiased inference strategy, MCCR-GNN maintains the best recommendation performance. We attribute this superiority to three aspects: 1) The constructed shared global embedding and the local embedding of the MC interaction graph can effectively capture the heterogeneity in user behavior; 2) The developed graph attention aggregation mechanism can effectively fuse the user's dependence on different criteria. 3) The designed self-supervised loss can effectively improve the robustness of user representation and item representation through cross-view knowledge propagation.
- The performance of the MCCR is further significantly improved compared to the MCCR-GNN. This is attributed to its property of mining causality according to the back-door adjustment during inference. After removing the causal intervention, MCCR-GNN achieves the prediction purpose based on the correlation modeling paradigm $P(R|U, I)$. We argue that this paradigm may limit the recommendation quality of the model due to spurious association caused by the back-door path. In contrast, the MCCR implements Top-N recommendation based on the causal modeling mechanism $P(R|do(U, I))$. This mechanism effectively mitigates the negative impact of spurious association on the model, thereby improving the decision accuracy of RSs.

5.2.2 Performance Analysis in Sparse Scenarios. Although MC ratings provide rich auxiliary information, this also significantly increases the sparsity of the data. To examine the performance of the MCCR in sparse scenarios, we attack the number of interactions in the overall rating matrix and the MC rating matrices. Specifically, we reduce the proportion of positive samples in the training set by randomly removing user-item interaction pairs. The sparser environment constructed by this approach can comprehensively evaluate the performance of the MCCR compared to existing baseline methods. We progressively remove the number of training samples at a rate of 10%, 20%, 30%, 40% and 50%, respectively.

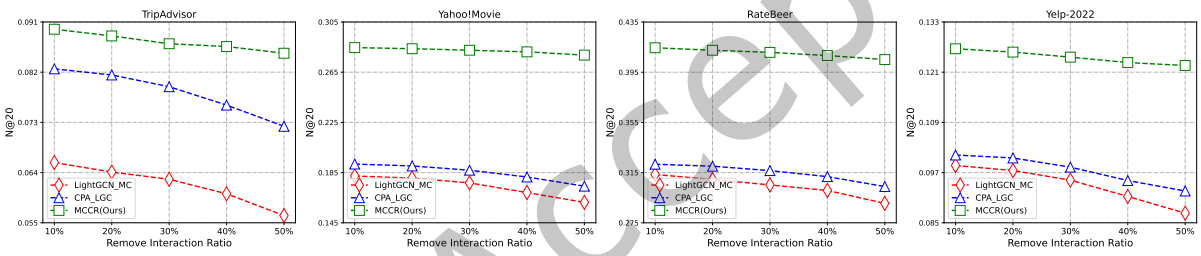
Figure 4 illustrates the comparison of Top-20 recommendation performance on the four MC datasets. Overall, the proposed MCCR consistently outperforms other baseline models in different sparse scenarios. It can be seen that as the proportion of positive samples in user interactions continues to decrease, the performance of each model shows a downward trend, but the decline speed of the MCCR appears more gentle compared to the other



(a) HR@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.



(b) Recall@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.



(c) NDCG@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.

Fig. 4. Top-N performance in sparse interaction scenarios.

baseline models. On the one hand, this phenomenon can be attributed to the unique design of the MCCR, in particular the contrast loss introduced among the overall rating view and the MC rating views. The optimization goal is to make the model more focused on learning stable and discriminative feature representations by enhancing the "positive" and "negative" contrasts among samples. In other words, this self-supervised optimization strategy can effectively filter out irrelevant interaction noise, thereby improving the robustness of the model in sparse scenarios. On the other hand, we argue that the superiority of the MCCR also stems from the contribution of causal inference to model prediction. The MCCR deeply mines the causal relationships between user behavior and recommendation decisions through the proposed causal graph. This prior knowledge based on causal inference reduces the model's dependence on data to a certain extent, thereby enhancing its generalization ability and enabling the model to make accurate predictions even in sparse environments.

5.3 Debiasing Performance (RQ2)

In this subsection, we evaluate the performance of the MCCR with the existing causal methods and compare the debiasing performance of the baseline methods on different backbone models.

5.3.1 Performance Comparison with the Debiasing Methods. We report the debiasing performance comparison of the MCCR with five causal methods in four MC scenarios. All methods employ the constructed MCCR-GNN as

Table 5. Debiasing performance comparison of MCCR and causal models on TripAdvisor and Yahoo!Movie.

Datasets	TripAdvisor						Yahoo!Movie					
	Top-20			Top-50			Top-20			Top-50		
Model	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
MCCR-GNN	0.1507	0.0764	0.0855	0.2021	0.1064	0.0981	0.5833	0.1918	0.2346	0.7669	0.3017	0.2657
IPW [50]	0.1503	0.0761	0.0852	0.2022	0.1063	0.0979	0.5867	0.1921	0.2355	0.7672	0.3038	0.2691
DR [32]	0.1506	0.0759	0.0857	0.2027	0.1065	0.0985	0.5871	0.1928	0.2358	0.7673	0.3044	0.2706
PDA [72]	0.1522	0.0775	0.0863	0.2055	0.1068	0.0989	0.6028	0.1944	0.2391	0.7734	0.3095	0.2881
DecRS [59]	0.1593	0.0806	0.0879	0.2109	0.1072	0.1006	0.6133	0.1998	0.2617	0.7861	0.3147	0.3122
DCF [64]	0.1588	0.0803	0.0881	0.2104	0.1075	0.1008	0.6127	0.1995	0.2602	0.7858	0.3151	0.3128
MCCR(Ours)	**0.1662***	**0.0837***	**0.0904***	**0.2166***	**0.1108***	**0.1025***	**0.6419***	**0.2066***	**0.2853***	**0.7981***	**0.3234***	**0.3593***
%improv.	4.33%	3.85%	2.61%	2.70%	3.07%	1.69%	4.66%	3.40%	9.02%	1.53%	2.63%	14.87%

The bold score denotes the best experimental result and the underlined score indicates the best baseline. %improv. denotes the relative improvement of MCCR compared to the best baseline. "*" denotes statistically significant improvement compared to the best baseline (p-value < 0.05).

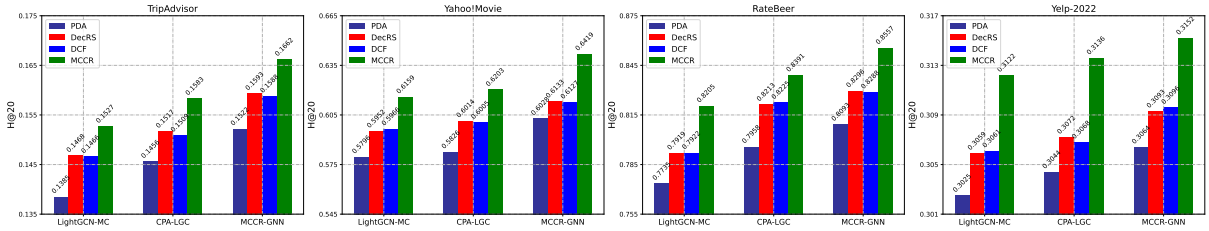
Table 6. Debiasing performance comparison of MCCR and causal models on RateBeer and Yelp-2022.

Datasets	RateBeer						Yelp-2022					
	Top-20			Top-50			Top-20			Top-50		
Model	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
MCCR-GNN	0.8012	0.3397	0.3661	0.8952	0.5128	0.4626	0.3004	0.1397	0.1123	0.4291	0.2335	0.1383
IPW [50]	0.8005	0.3382	0.3658	0.8937	0.5126	0.4625	0.3015	0.1403	0.1123	0.4293	0.2336	0.1389
DR [32]	0.8007	0.3385	0.3651	0.8939	0.5128	0.4631	0.3018	0.1402	0.1129	0.4295	0.2345	0.1411
PDA [72]	0.8093	0.3416	0.3716	0.9065	0.5174	0.4743	0.3064	0.1426	0.1165	0.4337	0.2381	0.1486
DecRS [59]	0.8296	0.3493	0.3918	0.9152	0.5227	0.4942	0.3093	0.1441	0.1201	0.4389	0.2407	0.1558
DCF [64]	0.8288	0.3491	0.3926	0.9156	0.5223	0.4943	0.3096	0.1443	0.1198	0.4392	0.2403	0.1552
MCCR(Ours)	**0.8557***	**0.3562***	**0.4166***	**0.9388***	**0.5309***	**0.5175***	**0.3152***	**0.1475***	**0.1275***	**0.4433***	**0.2456***	**0.1687***
%improv.	3.15%	1.98%	6.11%	2.53%	1.57%	4.69%	1.81%	2.22%	6.16%	0.93%	2.04%	8.28%

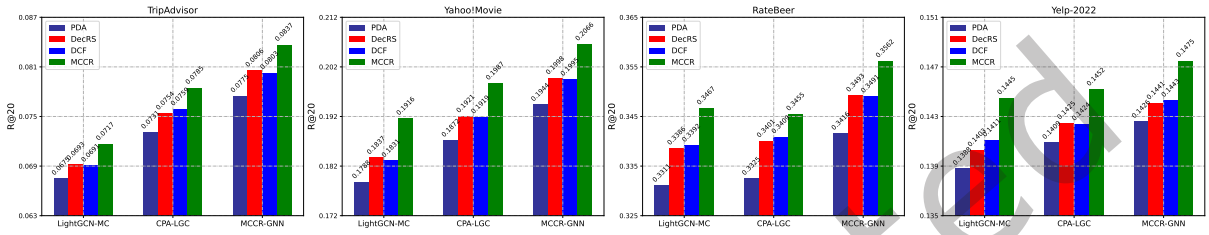
The bold score denotes the best experimental result and the underlined score indicates the best baseline. %improv. denotes the relative improvement of MCCR compared to the best baseline. "*" denotes statistically significant improvement compared to the best baseline (p-value < 0.05).

the backbone model. Tables 5 and 6 show the performance comparison on Top-20 and Top-50. We verify that the MCCR has a statistically significant improvement compared to the best baseline by a t-test (p-value < 0.05). We summarize the following conclusions from the experimental results:

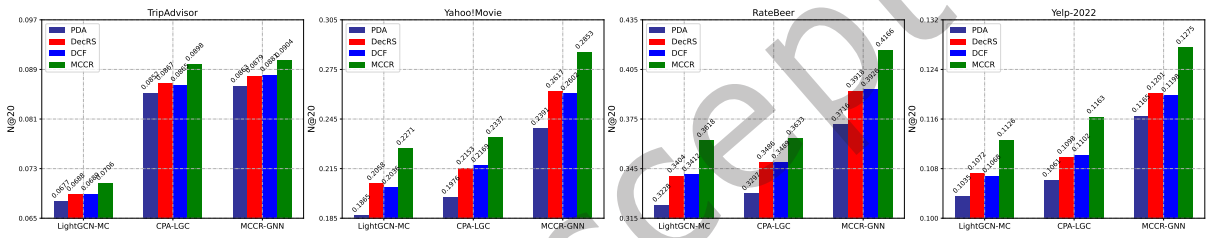
- MCCR consistently outperforms all baseline models. This superiority is attributed to its back-door adjustment strategy during inference, which alleviates the negative impact of bias on RSs through unbiased estimation. Compared to the best baseline, the MCCR improves by an average of 3.99% on all three metrics. These experimental results validate the rationality of causal analysis in MC recommendation scenarios and the necessity of employing causal interventions for debiasing.
- Both IPW and DR are classical debiasing methods, but they perform poorly on the four datasets. This phenomenon is mainly due to the sparsity of user feedback in MC scenarios, which causes them to face large variance when calculating weights. The instability of the weights limits their debiasing performance in the MC recommendation task. In addition, the high-order heterogeneity of the users' MC preferences results in IPW and DR becoming extremely challenging in capturing complex causal relationships.
- DecRS and DCF lead alternately and both outperform PDA. Since PDA is good at mitigating the negative impact of popularity bias on RSs, its performance is limited when dealing with the selection bias caused by the difference of users' MC interests. Unlike PDA, DecRS is suitable for solving the bias problem induced by the user preference distribution, DCF focuses on the bias caused by the confounders. Although DecRS and DCF achieve good performance, they ignore the higher-order user behavior information carried by MC ratings. Different from the existing causal methods, the proposed MCCR specifically deals with the bias problem in MC scenarios, which guarantees that the model achieves unbiased prediction in MC recommendation.



(a) HR@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.



(b) Recall@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.



(c) NDCG@20 on TripAdvisor, Yahoo!Movie, RateBeer, and Yelp-2022.

Fig. 5. Debiasing performance in MC recommendation scenarios.

5.3.2 Debiasing Performance on Different Backbone Models. To further validate the effectiveness of the MCCR in causal intervention, we select LightGCN-MC, CPA-LGC, and MCCR-GNN as the backbone models for the debiasing experiments. Specifically, we adopt PDA, DecRS, DCF, and MCCR for intervention inference in the prediction phase of each backbone model and compare their performance in four MC scenarios. Figure 5 illustrates the Top-20 recommendation performance of the four causal methods on the three backbone models. The experimental results again demonstrate the effectiveness of the proposed MCCR in mitigating the bias problem in MC scenarios. We argue that the superiority of the MCCR stems from the following two aspects:

- Causality mining. We leverage the SCM to deeply explore the causal relationships between user behavior and model predictions in MC scenarios from the perspective of data generation, which reveals the real reason for bias amplification. As discussed in Section 4.1.2, the backdoor path opened by confounders may generate spurious correlations, thereby degrading the predictive quality of the model. Therefore, clearly identifying and eliminating these spurious association is key to improving recommendation performance.
- Unique inference paradigm. The inference paradigm designed for MC rating recommendations employs the back-door adjustment to block the back-door path, which achieves unbiased estimation during model decision making. As mentioned in Section 4.1.3, the proposed inference strategy effectively removes the disturbance of confounders and enhances the accuracy of model prediction. Therefore, correcting the model to prevent bias amplification is important for exploiting complex user preferences.

Table 7. Ablation studies on TripAdvisor and Yahoo!Movie.

Datasets	TripAdvisor						Yahoo!Movie					
	Top-20			Top-50			Top-20			Top-50		
Variants	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
w/o MCR	0.1435	0.0716	0.0791	0.1856	0.0976	0.0912	0.5604	0.1863	0.2215	0.7206	0.2854	0.2357
w/o GNN	0.0864	0.0495	0.0312	0.1137	0.0628	0.0414	0.3622	0.1246	0.1638	0.4753	0.1968	0.1916
w/o GAT	0.1621	0.0802	0.0873	0.2116	0.1079	0.1012	0.6276	0.2014	0.2708	0.7735	0.3152	0.3263
w/o SSL	0.1615	0.0793	0.0866	0.2105	0.1051	0.0994	0.6232	0.1998	0.2631	0.7712	0.3143	0.3261
w/o BDA	0.1507	0.0764	0.0855	0.2021	0.1064	0.0981	0.5833	0.1918	0.2346	0.7669	0.3017	0.2657
MCCR	**0.1662**	**0.0837**	**0.0904**	**0.2166**	**0.1108**	**0.1025**	**0.6419**	**0.2066**	**0.2853**	**0.7981**	**0.3234**	**0.3593**

The bold scores indicate the best experimental results.

Table 8. Ablation studies on RateBeer and Yelp-2022.

Datasets	RateBeer						Yelp-2022					
	Top-20			Top-50			Top-20			Top-50		
Variants	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
w/o MCR	0.7713	0.3222	0.3565	0.8581	0.4767	0.4133	0.2869	0.1313	0.1027	0.4125	0.2158	0.1242
w/o GNN	0.4945	0.1728	0.2205	0.6737	0.3894	0.2547	0.1527	0.0973	0.0645	0.2386	0.0975	0.0892
w/o GAT	0.8369	0.3497	0.3968	0.9114	0.5207	0.5018	0.3078	0.1433	0.1192	0.4355	0.2394	0.1546
w/o SSL	0.8362	0.3476	0.3964	0.9105	0.5176	0.5004	0.3055	0.1429	0.1187	0.4352	0.2381	0.1535
w/o BDA	0.8012	0.3397	0.3661	0.8952	0.5128	0.4626	0.3004	0.1397	0.1123	0.4291	0.2335	0.1383
MCCR	**0.8557**	**0.3562**	**0.4166**	**0.9388**	**0.5309**	**0.5175**	**0.3152**	**0.1475**	**0.1275**	**0.4433**	**0.2456**	**0.1687**

The bold scores indicate the best experimental results.

5.4 Ablation Experiment (RQ3)

In this subsection, we evaluate the impact of different components of the MCCR on the model performance and the ablation studies for MC ratings.

5.4.1 Ablation Studies on Different Components of the MCCR. To validate the effectiveness of different components of the proposed MCCR on the recommendation performance, we design the following five variants of the MCCR:

- ***w/o MCR***: MC ratings are removed during modeling and only overall rating is retained for prediction. It should be noted that after removing MC ratings, the model's back-door adjustment strategy fails as the user's MC preference distribution cannot be computed.
- ***w/o GNN***: The constructed GNN architecture is removed and the user MC preferences are modeled with the Multi-Layer Perceptron.
- ***w/o GAT***: The coefficients used to measure the degree of association among the criteria are removed, and a simple average weighting is used instead of the graph attention network in MC information propagation.
- ***w/o SSL***: The self-supervised contrast loss is removed and the model parameters are updated based on the BPR optimization objective.
- ***w/o BDA***: The back-door adjustment strategy in the model inference phase is removed and the recommendation is implemented by adopting the traditional data-driven method.

Tables 7 and 8 report the results of the ablation studies on the four MC datasets. We can see that the performance of the variant "w/o MCR" decreases significantly after removing MC ratings. This indicates the importance of higher-order information in MC ratings for mining complex user preferences. The performance of the variant "w/o GNN" is the worst, which highlights the key role of GNN in capturing higher-order associations in user interaction data. Compared with traditional Multi-Layer Perceptron, the GNN-specific topology aggregation mechanism can model user preferences more adequately, especially in multiple sparse MC views. Although the performance of variant "w/o GAT" is relatively good, it still does not reach the level of the MCCR, which is attributed to the fact that simple average weighting fails to effectively reflect the degree of user preference for different criteria. In other words, effectively extracting the heterogeneity of user preferences is crucial to

Table 9. Ablation studies on MC ratings.

Datasets	Variants	Top-20				Top-50			
		H@20	R@20	N@20	%DR.	H@50	R@50	N@50	%DR.
TripAdvisor	w/o R1	0.1491	0.0746	0.0836	-1.88%	0.1965	0.1052	0.0958	-2.08%
	w/o R3	0.1468	0.0731	0.0815	-3.86%	0.1911	0.1027	0.0936	-4.50%
	w/o R5	0.1451	0.0718	0.0792	-5.70%	0.1878	0.0983	0.0915	-7.14%
	MCCR-GNN	**0.1507**	**0.0764**	**0.0855**	-	**0.2021**	**0.1064**	**0.0981**	-
	w/o R1	0.1613	0.0819	0.0894	-2.07%	0.2134	0.1095	0.0987	-2.12%
	w/o R3	0.1574	0.0812	0.0863	-4.27%	0.2083	0.1037	0.0971	-5.17%
	w/o R5	0.1535	0.0795	0.0855	-6.03%	0.1977	0.0996	0.0946	-8.85%
	MCCR	**0.1662**	**0.0837**	**0.0904**	-	**0.2166**	**0.1108**	**0.1025**	-
Yahoo!Movie	w/o R1	0.5787	0.1905	0.2313	-0.96%	0.7585	0.2931	0.2628	-1.68%
	w/o R2	0.5724	0.1883	0.2294	-1.97%	0.7422	0.2883	0.2561	-3.76%
	w/o R3	0.5662	0.1871	0.2236	-3.36%	0.7293	0.2857	0.2417	-6.41%
	MCCR-GNN	**0.5833**	**0.1918**	**0.2346**	-	**0.7669**	**0.3017**	**0.2657**	-
	w/o R1	0.6361	0.2058	0.2829	-0.71%	0.7864	0.3206	0.3518	-1.47%
	w/o R2	0.6259	0.2014	0.2776	-2.57%	0.7743	0.3155	0.3482	-2.84%
	w/o R3	0.6175	0.2001	0.2715	-3.93%	0.7721	0.3093	0.3313	-5.14%
	MCCR	**0.6419**	**0.2066**	**0.2853**	-	**0.7981**	**0.3234**	**0.3593**	-
RateBeer	w/o R1	0.7813	0.3324	0.3634	-1.79%	0.8837	0.5003	0.4526	-1.96%
	w/o R2	0.7779	0.3286	0.3602	-2.60%	0.8765	0.4966	0.4438	-3.10%
	w/o R3	0.7718	0.3259	0.3587	-3.25%	0.8682	0.4824	0.4216	-5.94%
	MCCR-GNN	**0.8012**	**0.3397**	**0.3661**	-	**0.8952**	**0.5128**	**0.4626**	-
	w/o R1	0.8434	0.3467	0.4151	-1.49%	0.9212	0.5234	0.5033	-2.01%
	w/o R2	0.8368	0.3416	0.4109	-2.56%	0.9167	0.5171	0.4865	-3.65%
	w/o R3	0.8261	0.3375	0.4053	-3.81%	0.9058	0.4923	0.4771	-6.20%
	MCCR	**0.8557**	**0.3562**	**0.4166**	-	**0.9388**	**0.5309**	**0.5175**	-
Yelp-2022	w/o R1	0.2959	0.1361	0.1052	-3.47%	0.4212	0.2267	0.1312	-3.30%
	w/o R2	0.2903	0.1314	0.1049	-5.30%	0.4176	0.2179	0.1266	-5.94%
	MCCR-GNN	**0.3004**	**0.1397**	**0.1123**	-	**0.4291**	**0.2335**	**0.1383**	-
	w/o R1	0.3102	0.1418	0.1208	-3.57%	0.4363	0.2365	0.1628	-2.93%
	w/o R2	0.3016	0.1377	0.1195	-5.74%	0.4255	0.2297	0.1534	-6.52%
	MCCR	**0.3152**	**0.1475**	**0.1275**	-	**0.4433**	**0.2456**	**0.1687**	-

The bold scores indicate the best experimental results and "%DR." indicates the average decline rate.

the performance improvement of the model. MCCR adaptively learns attention coefficients among criteria to capture the importance of each criterion for individual users. The performance of the variant "w/o SSL" and the variant "w/o GAT" is similar, but slightly decreased on multiple metrics, which emphasizes the positive role of self-supervised contrast loss in facilitating model to learn robust and discriminative feature representations. The contrast loss among criteria enhances the representation learning of sparse features by narrowing the embedding distance of the same user on different views. Compared to the MCCR, the performance of the variant "w/o BDA" is significantly reduced, which validates the necessity of causal intervention to mitigate the negative impact of confounders on RSs. In summary, the results of the ablation studies clearly demonstrate that each component of the MCCR has a positive effect on modeling users' MC preferences, which is consistent with our previous theoretical analysis.

5.4.2 Ablation Studies on MC Ratings. To validate the effectiveness of MC ratings in improving recommendation performance, we design variant "w/o RN" based on the MCCR-GNN and the MCCR, which randomly removes N criteria. Specifically, we randomly remove 1, 3, and 5 criterion interaction graphs on TripAdvisor, respectively; We randomly remove 1, 2, and 3 criterion views on Yahoo!Movie and RateBeer, respectively; We randomly remove 1 and 2 views on Yelp-2022, respectively. The experimental results are shown in Table 9, where %improv. denotes the average performance degradation on the three metrics. It can be seen that the rate of model performance degradation increases significantly as more MC ratings are removed. These results prove that MC ratings play an important role in boosting RSs. The additional auxiliary information enhances the understanding of user

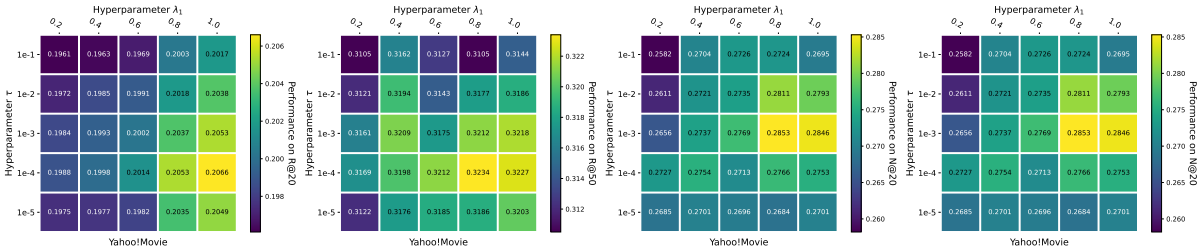


Fig. 6. Hyperparameter sensitivity w.r.t. λ_1 and τ on Yahoo!Movie.

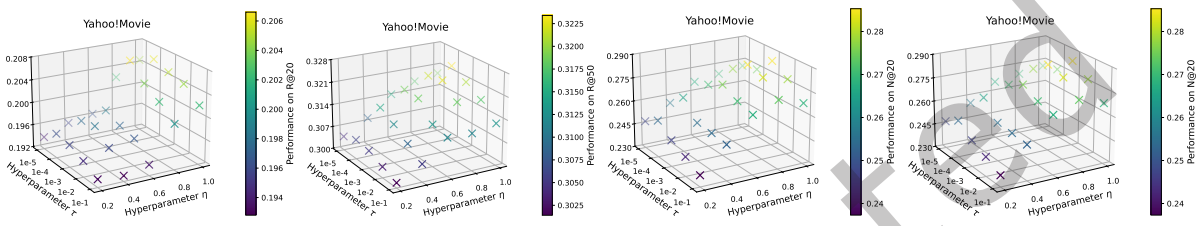


Fig. 7. Hyperparameter sensitivity w.r.t. η and τ on Yahoo!Movie.

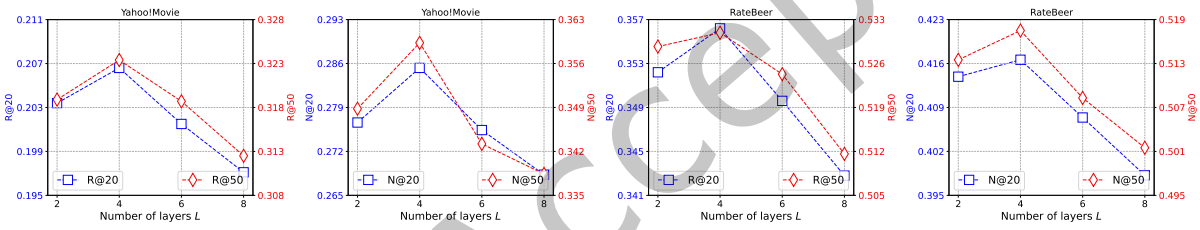


Fig. 8. Hyperparameter sensitivity w.r.t. L on Yahoo!Movie and RateBeer.

interaction behavior, thereby improving the model's ability to explore diverse user preferences. Therefore, developing a recommendation framework suitable for MC rating is a reasonable research motivation.

5.5 Hyperparametric Sensitivity Analysis (RQ4)

In this subsection, we evaluate the impact of four hyperparameters of the MCCR on the recommendation performance: 1) the self-supervised loss coefficient λ_1; 2) the BPR loss coefficient η on the MC ratings; 3) the temperature coefficient τ; and 4) the number of GNN layers L.

5.5.1 Sensitivity Analysis of the Self-Supervised Loss Coefficient λ_1. The hyperparameter λ_1 is used to control the trade-off between the self-supervised contrast loss $\mathcal{L}_{cl}$ and the main recommendation task loss $\mathcal{L}_{bpr}$ in the MCCR model. This parameter is crucial for improving the robustness and discrimination of the model. Specifically, the hyperparameter λ_1 learns more discriminative features by narrowing similar embeddings, such as users or items with similar MC ratings, and distancing dissimilar embeddings. In this paper, λ_1 is tuned to be in range $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. Figure 6 illustrates the impact of λ_1 on Yahoo!Movie. It can be seen that the model performance is optimized when λ_1 is around 0.8. The decrease of λ_1 value weakens the effect of contrast strength, which results in insufficient discrimination of the model embedding and a decline of recommendation performance. When the value of λ_1 is close to 1, we guess that this may lead to an optimization imbalance between the contrastive loss and the main task loss, making the recommendation quality worse. Overall, a moderate λ_1

value can effectively balance the self-supervised and recommendation objective to achieve the best performance on MC data.

5.5.2 Sensitivity Analysis of the BPR Loss Coefficient η for MC Ratings. The hyperparameter η is used to regulate the optimized strength of the BPR loss based on MC ratings in the MCCR. This parameter affects the model's ability to capture the user's personalized preferences by assigning different weights to the MC rating loss. In this paper, the value of η is set in the range $\{0.2, 0.4, 0.6, 0.8, 1.0\}$. Figure 7 reports the performance impact of η on yahoo. It can be seen that when the η value is low, the model is unable to fully utilize the feedback from the MC ratings, resulting in poor performance in capturing diverse user preferences. This is attributed to the fact that the model may focus more on the overall rating and cannot fully exploit the higher-order information provided by the MC ratings. It is worth noting that when $\eta = 1$, the model does not reach the optimum in all metrics. We speculate that this may be due to overfitting causing the model to be overly sensitive to subtle differences in certain criteria. Therefore, setting a reasonable η value can help the model capture complex user-item interactions and improve recommendation quality.

5.5.3 Sensitivity Analysis of the Temperature Coefficient τ. The hyperparameter τ is used to control the sensitivity of the embedding features in similarity computation among the overall rating view and the MC rating views. In this paper, we adjust τ in range $\{1e-5, 1e-4, 1e-3, 1e-2, 1e-1\}$. Figures 6 and 7 show the impact of different values of τ on the MCCR. The experimental results show that the performance of the model is optimized when the τ value is 0.2 or 0.3. We argue that too small τ value will greatly approximate the embedding distance of similar samples, which may result in the overfitting problem and reduce the generalization ability of the model. When the value of τ is too large, the role of contrast loss is weakened, which may cause the model cannot effectively distinguish the differences in users' MC behavioral characteristics. Therefore, setting a reasonable τ value can help the model to improve the learning capability of the embedding representation and enhance the recommendation accuracy.

5.5.4 Sensitivity Analysis of the GNN Layers L. The number of GNN layers L determines the ability of the MCCR to capture higher-order relationships in user-item interaction graphs. In this paper, L is tuned in range $\{2, 4, 6, 8\}$. Figure 8 illustrates the performance impact of different L values on Yahoo!Movie and RateBeer. We can observe that the model performance is optimal when $L = 4$. Too low or too high number of layers can weaken the predictive performance of the model. We believe that too small L limits the ability of the model to aggregate higher-order features in the graph structure. Due to the lack of deep dependency information in the embedding representation, it is difficult for the model to capture the complexity of the user's preferences. In addition, although increasing the number of layers allows the model to integrate features from more distant neighbors, this may cause the over-smoothing problem. In this case, the node representations tend to be similar and cannot reflect the personalized needs of users. In other words, a large L makes the feature fusion between different nodes of the model too uniform and weakens the expression ability of different information. Therefore, choosing an appropriate L value to ensure that the MCCR achieves a balance between capturing local and global interaction patterns is key to improving the decision-making accuracy of the model.

5.6 Exploratory Analysis

To further explore MCCR, we design performance comparison on large datasets, case study, and efficiency analysis.

5.6.1 Performance Comparison on Large Datasets. To evaluate the performance of the proposed MCCR in larger scale recommendation scenarios, we conduct experiments on two large datasets. The statistical information of the datasets is shown in Table 10, specifically:

Table 10. Dataset statistics. #Overall and #MC denote the interaction numbers of overall rating and MC ratings, respectively, and K denotes the number of criteria.

Dataset	#Users	#Items	#Overall	#MC	K	Sparsity
BeerAdvocate	33,388	66,055	1,586,614	6,346,442	4	8.99E-04
RB-Extended	40,213	110,419	2,924,163	11,696,652	4	8.23E-04

Table 11. Performance comparison of MCCR and data-driven models on BeerAdvocate and RB-Extended.

Datasets	BeerAdvocate						RB-Extended					
	Top-20			Top-50			Top-20			Top-50		
Model	H@20	R@20	N@20	H@50	R@50	N@50	H@20	R@20	N@20	H@50	R@50	N@50
GC-MC [7]	0.3254	0.0863	0.1088	0.4352	0.1407	0.1435	0.3952	0.1013	0.1476	0.4804	0.1687	0.1895
SpectralCF [76]	0.2961	0.0737	0.0976	0.4134	0.1289	0.1324	0.3917	0.0988	0.1405	0.4629	0.1558	0.1767
NGCF [61]	0.3536	0.0904	0.1125	0.4481	0.1539	0.1563	0.3946	0.1008	0.1453	0.4768	0.1643	0.1822
DGCF [62]	0.3573	0.0925	0.1193	0.4492	0.1527	0.1588	0.3759	0.0914	0.1282	0.4425	0.1506	0.1615
LightGCN [24]	0.3658	0.0976	0.1282	0.4655	0.1603	0.1625	0.3966	0.1005	0.1489	0.4793	0.1694	0.1871
UBM [77]	0.1974	0.0482	0.0617	0.2563	0.0719	0.0723	0.2129	0.0638	0.0817	0.2673	0.0984	0.1126
DMCF [42]	0.2166	0.0613	0.0755	0.2964	0.0897	0.0951	0.2567	0.0696	0.0885	0.2914	0.1038	0.1197
AEMC [51]	0.2539	0.0798	0.0846	0.3802	0.1056	0.1279	0.3252	0.0814	0.0895	0.3681	0.1159	0.1268
CFM [13]	0.3145	0.0911	0.1075	0.4296	0.1322	0.1418	0.3771	0.0905	0.1092	0.4253	0.1306	0.1467
LightGCN_MC [24]	0.3764	0.1007	0.1328	0.4751	0.1696	0.1703	0.4068	0.1027	0.1543	0.4918	0.1773	0.1965
CPA-LGC [44]	0.3855	0.1016	0.1356	0.4838	0.1762	0.1787	0.4128	0.1054	0.1602	0.5037	0.1815	0.2016
MCCR-GNN	0.3908	0.1022	0.1516	0.4961	0.1794	0.2168	0.4193	0.1085	0.1807	0.5162	0.1867	0.2469
MCCR(Ours)	**0.4122***	**0.1074***	**0.1749***	**0.5187***	**0.1869***	**0.2422***	**0.4486***	**0.1141***	**0.2063***	**0.5524***	**0.1968***	**0.2798***
%improv.	6.93%	5.71%	28.98%	7.21%	6.07%	35.53%	8.67%	8.25%	28.78%	9.67%	8.43%	38.79%

The bold score denotes the best experimental result and the underlined score indicates the best baseline. %improv. denotes the relative improvement of MCCR compared to the best baseline. "*" denotes statistically significant improvement compared to the best baseline (p-value < 0.01).

- **BeerAdvocate**[5] is a rating data about the beer and the criteria include *appearance*, *palate*, *aroma* and *taste* on a scale of 1 to 5.
- **RB-Extended**[3] is an expanded version of RateBeer, containing more extensive rating data under the same criteria. We name this dataset RB-Extended.

Table 11 reports the performance comparisons of various baselines. We can observe that the proposed model shows significant superiority on Top-20 and Top-50 recommendations. In particular, compared to the state-of-the-art baseline, the N@50 of MCCR on BeerAdvocate and RB-Extended increased by 35.53% and 38.79% respectively. This success is attributed to two aspects: 1) The constructed GNN architecture efficiently mines higher-order heterogeneous interactions in MC ratings and improves the quality of embedding representations based on the cross-criteria contractive learning mechanism; 2) The developed inference strategy adopts back-door adjustment to block the negative impact of confounding, thereby estimating unbiased user preferences. These results verify the effectiveness of MCCR on large-scale datasets and its application potential in real MC recommendation scenarios.

5.6.2 Case Study. To illustrate the debiasing performance of MCCR, we analyze the recommendation results of several models on Yelp-2022. Specifically, we select three typical users (IDs #1358, #3909, and #6375, respectively) whose criterion preferences have significant differences in the training set. Next, we count the recommendation lists predicted by CPA-LGC, MCCR-GNN, and MCCR, and calculate their item distributions on different criteria. We categorize items based on the historical rating percentage on the different criterion views.

It can be observed from Figure 9 that the highest interaction ratios of the *cool*, *funny* and *useful* views in the training set reach 58%, 51% and 53% respectively. We can find from the experimental results that: 1) CPA-LGC and MCCR-GNN exacerbate the distributional bias inherent in the data. This is because the data-driven learning paradigm achieves prediction by capturing correlations of user behavior. This paradigm may cause the model to overly recommend high-frequency interaction items due to the feedback loop; 2) MCCR alleviates the data bias

[5]https://www.beeradvocate.com/

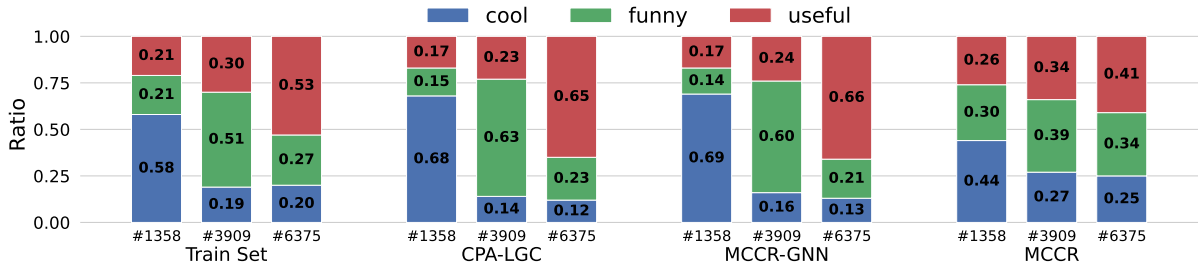


Fig. 9. A case study of the recommendation results on Yelp-2022.

Table 12. Efficiency comparisons between MCCR and baselines.

Dataset		Yahoo!Movie			RateBeer		
Model		Avg. Time	Tot. Time	# Epochs	Avg. Time	Tot. Time	# Epochs
MC Methods	DMCF	20.3 s	63.6 m	188	93.6 s	333.8 m	214
	CPA-LGC	13.7 s	28.3 m	124	75.8 s	200.9 m	159
	MCCR-GNN	12.8 s	20.5 m	96	62.4 s	138.3 m	133
Debiasing Methods	DecRS	13.9 s	26.6 m	115	66.5 s	160.7 m	145
	DCF	15.3 s	34.7 m	136	69.7 s	192.8 m	166
	MCCR	13.1 s	22.3 m	102	63.2 s	142.2 m	135

Runtime comparison (seconds/minutes [s/m]), including average time (Avg. Time) for each epoch, total time (Tot. Time), and the number of convergent epochs (# Epochs).

in the recommendation results. This indicates the rationality of the proposed causal modeling framework. This framework effectively broadens the user's perspective and suppresses the negative impacts of homogenization such as filter bubbles and information cocoons.

5.6.3 Efficiency Analysis. To validate the computational efficiency superiority of the proposed framework, we compare the runtime of MCCR with several baselines, including MC methods and debiasing methods. It is worth mentioning that the network backbone of the debiasing methods employs the MCCR-GNN designed in this paper. To ensure a fair comparison, all experiments are conducted under the same experimental conditions. Table 12 reports the efficiency results on Yahoo!Movie and RateBeer. We can observe that: 1) CPA-LGC and MCCR-GNN achieve competitive running efficiency compared to DMCF in the MC methods. This indicates that the inherent topology-aware property of graph neural networks compensates for the computational bottleneck faced by traditional neural networks when processing high-dimensional and sparse MC rating data. 2) Among the debiasing methods, MCCR has the least influence on the backbone model. For example, the average training time per epoch demonstrates that MCCR introduces almost no additional computational overhead. This can be attributed to the proposed inference strategy avoids traversing all item pools when estimating the probability $P(R|do(U,I))$. This strategy greatly reduces the computational cost of implementing unbiased estimation with back-door adjustment. In summary, the computational efficiency advantage of MCCR is consistent with our theoretical analysis.

6 CONCLUSION AND FUTURE WORK

In this work, we propose a novel MCCR recommendation framework for mitigating bias, which models the causal relationships between user behavior and recommendation decisions through the causal intervention. We also exploit the heterogeneity of user MC preferences by using graph convolution operation. Experimental results demonstrate that the proposed framework exhibits superior performance on six MC scenarios compared to the

existing baselines. Different from the existing MC methods, the MCCR has several advantages. First, the MCCR analyzes the reason why the bias problem is amplified by using causal inference. The MCCR cuts off the spurious association induced by confounding with the back-door adjustment, which improves the accuracy of RSs. To the best of our knowledge, this is the first attempt in the MC recommendation methods. Second, the constructed training and inference paradigm is model-independent, which improves the accuracy of RSs by formulating recommendation strategies through unbiased estimation. Third, the proposed architecture introduces GNN to extract high-order heterogeneous MC ratings and uses the graph attention mechanism to model the user's MC preferences. Fourth, the MCCR introduces a self-supervised contrastive loss as a complement to the cost function, which helps the model adapt to sparse interaction environments and improves the robustness of RSs.

In future work, several limitations need to be improved. For example, the designed static M may not accurately reflect the real-time transfer of user interests in practical applications [60], and how to design a dynamic framework to capture the changing user preferences is a crucial challenge. In addition, MC ratings may result in the optimization imbalance problem during the training process, so that the weights of the neural network are dominated by the MC information. How to design an adaptive optimization method for gradient updating is another important challenge. On the other hand, the MCCR has the limitation of scalability when faced with spurious correlation problems caused by unobservable variables. It is a reasonable and practical solution to address the above challenge by employing the front-door criterion.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 72171137, U21A20473, 62406180), and the Fundamental Research Program of Shanxi Province (No. 202203021211331, 202403021212337).

REFERENCES

[1] Gediminas Adomavicius and YoungOk Kwon. 2007. New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems* 22, 3 (2007), 48–55.

[2] Gediminas Adomavicius, Nikos Manouselis, and YoungOk Kwon. 2010. Multi-criteria recommender systems. In *Recommender systems handbook*. Springer, 769–803.

[3] M Mehdi Afsar, Trafford Crump, and Behrouz Far. 2022. Reinforcement learning based recommender systems: A survey. *Comput. Surveys* 55, 7 (2022), 1–38.

[4] Milad Ahmadian, Sajad Ahmadian, and Mahmood Ahmadi. 2023. RDERL: Reliable deep ensemble reinforcement learning-based recommender system. *Knowledge-Based Systems* 263 (2023), 110289.

[5] Kristen M Altenburger and Daniel E Ho. 2019. Is Yelp actually cleaning up the restaurant industry? A re-analysis on the relative usefulness of consumer reviews. In *The World Wide Web Conference*. 2543–2550.

[6] Keqin Bao, Jizhi Zhang, Yang Zhang, Wang Wenjie, Fuli Feng, and Xiangnan He. 2023. Large language models for recommendation: Progresses and future directions. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 306–309.

[7] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[8] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 19–26.

[9] Gang Chen, Jiawei Chen, Fuli Feng, Sheng Zhou, and Xiangnan He. 2023. Unbiased knowledge distillation for recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 976–984.

[10] 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*. 21–30.

[11] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[12] Juan D Correa, Jin Tian, and Elias Bareinboim. 2019. Identification of causal effects in the presence of selection bias. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2744–2751.

[13] Ge Fan, Chaoyun Zhang, Junyang Chen, and Kaishun Wu. 2021. Predicting ratings in multi-criteria recommender systems via a collective factor model. In *DeMal@ The Web Conference*. 1–6.
[14] Shaohua Fan, Xiao Wang, Chuan Shi, Peng Cui, and Bai Wang. 2024. Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, 1 (2024), 322–337.
[15] Shaohua Fan, Xiao Wang, Chuan Shi, Kun Kuang, Nian Liu, and Bai Wang. 2024. Debiased graph neural networks with agnostic label selection bias. *IEEE Transactions on Neural Networks and Learning Systems* 35, 4 (2024), 4411–4422.
[16] Wenqi Fan, Yao Ma, Qing Li, Jianping Wang, Guoyong Cai, Jiliang Tang, and Dawei Yin. 2020. A graph neural network framework for social recommendations. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2020), 2033–2047.
[17] Chenjiao Feng, Jiye Liang, Peng Song, and Zhiqiang Wang. 2020. A fusion collaborative filtering method for sparse data in recommender systems. *Information Sciences* 521 (2020), 365–379.
[18] Chaofan Fu, Guanjie Zheng, Chao Huang, Yanwei Yu, and Junyu Dong. 2023. Multiplex heterogeneous graph neural network with behavior pattern modeling. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 482–494.
[19] Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 238–248.
[20] Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023. CIRS: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems* 42, 1 (2023), 1–27.
[21] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2024. Causal inference in recommender systems: A survey and future directions. *ACM Transactions on Information Systems* 42, 4 (2024), 1–32.
[22] Shuyun Gu, Xiao Wang, Chuan Shi, and Ding Xiao. 2022. Self-supervised graph neural networks for multi-behavior recommendation. In *International Joint Conference on Artificial Intelligence*. 2052–2058.
[23] Huifeng Guo, Jinkai Yu, Qing Liu, Ruiming Tang, and Yuzhou Zhang. 2019. PAL: a position-bias aware learning framework for CTR prediction in live recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 452–456.
[24] 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.
[25] 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.
[26] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 549–558.
[27] Xiangnan He, Yang Zhang, Fuli Feng, Chonggang Song, Lingling Yi, Guohui Ling, and Yongdong Zhang. 2023. Addressing confounding feature issue for causal recommendation. *ACM Transactions on Information Systems* 41, 3 (2023), 1–23.
[28] Yuan He, Cheng Wang, and Changjun Jiang. 2018. Correlated matrix factorization for recommendation with implicit feedback. *IEEE Transactions on Knowledge and Data Engineering* 31, 3 (2018), 451–464.
[29] Yuheng Hu and Yili Hong. 2022. SHEDR: an end-to-end deep neural event detection and recommendation framework for hyperlocal news using social media. *INFORMS Journal on Computing* 34, 2 (2022), 790–806.
[30] Dietmar Jannach, Zeynep Karakaya, and Fatih Gedikli. 2012. Accuracy improvements for multi-criteria recommender systems. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 674–689.
[31] Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. 2010. *Recommender systems: an introduction*. Cambridge University Press.
[32] Nan Jiang and Lihong Li. 2016. Doubly robust off-policy value evaluation for reinforcement learning. In *International Conference on Machine Learning*. PMLR, 652–661.
[33] Kleanthi Lakiotaki, Nikolaos F Matsatsinis, and Alexis Tsoukias. 2011. Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems* 26, 2 (2011), 64–76.
[34] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.
[35] Jianing Li, Chaoqun Yang, Guanhua Ye, and Quoc Viet Hung Nguyen. 2024. Graph neural networks with deep mutual learning for designing multi-modal recommendation systems. *Information Sciences* 654 (2024), 119815.
[36] Pan Li and Alexander Tuzhilin. 2019. Latent multi-criteria ratings for recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 428–431.
[37] Pan Li and Alexander Tuzhilin. 2020. Learning latent multi-criteria ratings from user reviews for recommendations. *IEEE Transactions on Knowledge and Data Engineering* 34, 8 (2020), 3854–3866.
[38] Qiudan Li, Chunheng Wang, and Guanggang Geng. 2008. Improving personalized services in mobile commerce by a novel multicriteria rating approach. In *Proceedings of the 17th International Conference on World Wide Web*. 1235–1236.

[39] Haochen Liu, Da Tang, Ji Yang, Xiangyu Zhao, Hui Liu, Jiliang Tang, and Youlong Cheng. 2022. Rating distribution calibration for selection bias mitigation in recommendations. In *Proceedings of the ACM Web Conference 2022*. 2048–2057.
[40] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and S Yu Philip. 2022. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering* 35, 6 (2022), 5879–5900.
[41] Nikos Manouselis and Constantina Costopoulou. 2007. Analysis and classification of multi-criteria recommender systems. *World Wide Web* 10 (2007), 415–441.
[42] Nour Nassar, Assef Jafar, and Yasser Rahhal. 2020. A novel deep multi-criteria collaborative filtering model for recommendation system. *Knowledge-Based Systems* 187 (2020), 104811.
[43] Mehrbakhsh Nilashi, Dietmar Jannach, Othman bin Ibrahim, and Norafida Ithnin. 2015. Clustering-and regression-based multi-criteria collaborative filtering with incremental updates. *Information Sciences* 293 (2015), 235–250.
[44] Jin-Duk Park, Siqing Li, Xin Cao, and Won-Yong Shin. 2023. Criteria Tell You More than Ratings: Criteria Preference-Aware Light Graph Convolution for Effective Multi-Criteria Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1808–1819.
[45] Judea Pearl. 2009. *Causality*. Cambridge University Press.
[46] Yifang Qin, Wei Ju, Hongjun Wu, Xiao Luo, and Ming Zhang. 2024. Learning graph ODE for continuous-time sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3224–3236.
[47] Donald B Rubin. 1974. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology* 66, 5 (1974), 688.
[48] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 501–509.
[49] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*. 285–295.
[50] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning*. PMLR, 1670–1679.
[51] Qusai Shambour. 2021. A deep learning based algorithm for multi-criteria recommender systems. *Knowledge-Based Systems* 211 (2021), 106545.
[52] Jie Shuai, Le Wu, Kun Zhang, Peijie Sun, Richang Hong, and Meng Wang. 2023. Topic-enhanced graph neural networks for extraction-based explainable recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1188–1197.
[53] Zihua Si, Zhongxiang Sun, Xiao Zhang, Jun Xu, Yang Song, Xiaoxue Zang, and Ji-Rong Wen. 2023. Enhancing recommendation with search data in a causal learning manner. *ACM Transactions on Information Systems* 41, 4 (2023), 1–31.
[54] Zijie Song, Jiawei Chen, Sheng Zhou, Qihao Shi, Yan Feng, Chun Chen, and Can Wang. 2023. CDR: Conservative doubly robust learning for debiased recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 2321–2330.
[55] Dharahas Tallapally, Rama Syamala Sreepada, Bidyut Kr Patra, and Korra Sathya Babu. 2018. User preference learning in multi-criteria recommendations using stacked auto encoders. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 475–479.
[56] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, Yoshua Bengio, et al. 2017. Graph attention networks. *Stat* 1050, 20 (2017), 10–48550.
[57] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1235–1244.
[58] Shuyao Wang, Yongduo Sui, Chao Wang, and Hui Xiong. 2024. Unleashing the Power of Knowledge Graph for Recommendation via Invariant Learning. In *Proceedings of the ACM on Web Conference 2024*. 3745–3755.
[59] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1717–1725.
[60] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal representation learning for out-of-distribution recommendation. In *Proceedings of the ACM Web Conference 2022*. 3562–3571.
[61] 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.
[62] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1001–1010.
[63] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2024. Counterfactual explanation for fairness in recommendation. *ACM Transactions on Information Systems* 42, 4 (2024), 1–30.
[64] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2020. Causal inference for recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*. 426–431.

[65] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems* 41, 3 (2023), 1–43.
[66] Yinwei Wei, Xiang Wang, Liqiang Nie, Shaoyu Li, Dingxian Wang, and Tat-Seng Chua. 2023. Causal inference for knowledge graph based recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 11 (2023), 11153–11164.
[67] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 5 (2022), 4425–4445.
[68] Yonghui Yang, Le Wu, Zihan Wang, Zhuangzhuang He, Richang Hong, and Meng Wang. 2024. Graph Bottlenecked Social Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3853–3862.
[69] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-contrastive graph learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1117–1126.
[70] Eva Zangerle and Christine Bauer. 2022. Evaluating recommender systems: survey and framework. *Comput. Surveys* 55, 8 (2022), 1–38.
[71] Jizhi Zhang, Keqin Bao, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Is chatgpt fair for recommendation? evaluating fairness in large language model recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 993–999.
[72] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 11–20.
[73] Yan Zhang, Defu Lian, and Guowu Yang. 2017. Discrete personalized ranking for fast collaborative filtering from implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 1669–1675.
[74] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2022. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).
[75] Zhongying Zhao, Zhan Yang, Chao Li, Qingtian Zeng, Weili Guan, and MengChu Zhou. 2022. Dual feature interaction-based graph convolutional network. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 9019–9030.
[76] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S Yu. 2018. Spectral collaborative filtering. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 311–319.
[77] Yong Zheng. 2019. Utility-based multi-criteria recommender systems. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. 2529–2531.
[78] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.