Revisiting Reciprocal Recommender Systems: Metrics, Formulation, and Method

Chen Yang
flust@ruc.edu.cn
Nanbeige Lab, BOSS Zhipin
Beijing, China
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing, China

Sunhao Dai
sunhaodai@ruc.edu.cn
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing, China

Yupeng Hou
yphou@ucsd.edu.cn
University of California, San Diego
La Jolla, United States

Wayne Xin Zhao ✉
batmanfly@gmail.com
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing, China

Jun Xu
junxu@ruc.edu.cn
Gaoling School of Artificial
Intelligence, Renmin University of
China
Beijing, China

Yang Song ✉
songyang@kanzhun.com
Nanbeige Lab, BOSS Zhipin
Beijing, China

Hengshu Zhu
zhuhengshu@kanzhun.com
Career Science Lab, BOSS Zhipin
Beijing, China

Abstract

Reciprocal recommender systems (RRS), conducting bilateral recommendations between two involved parties, have gained increasing attention for enhancing matching efficiency. However, the majority of existing methods in the literature still reuse conventional ranking metrics to separately assess the performance on each side of the recommendation process. These methods overlook the fact that the ranking outcomes of both sides collectively influence the effectiveness of the RRS, neglecting the necessity of a more holistic evaluation and a capable systemic solution.

In this paper, we systemically revisit the task of reciprocal recommendation, by introducing the new metrics, formulation, and method. Firstly, we propose five new evaluation metrics that comprehensively and accurately assess the performance of RRS from three distinct perspectives: overall coverage, bilateral stability, and balanced ranking. These metrics provide a more holistic understanding of the system's effectiveness and enable a comprehensive evaluation. Furthermore, we formulate the RRS from a causal

perspective, formulating recommendations as bilateral interventions, which can better model the decoupled effects of potential influencing factors. By utilizing the potential outcome framework, we further develop a model-agnostic causal reciprocal recommendation method that considers the causal effects of recommendations. Additionally, we introduce a reranking strategy to maximize matching outcomes, as measured by the proposed metrics. Extensive experiments on two real-world datasets from recruitment and dating scenarios demonstrate the effectiveness of our proposed metrics and approach. The code and dataset are available at: https://github.com/RUCAIBox/CRRS.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Reciprocal Recommendation, Evaluation Metrics, Causal Inference

✉ Corresponding author.

1 Introduction

Reciprocal recommender systems (RRS), refer to the task that two-sided adoption has to be achieved for the final match, which has been widely studied and applied to scenarios like marriage markets [5, 14, 24, 41], recruitment [45, 47], and social networks [1, 9].
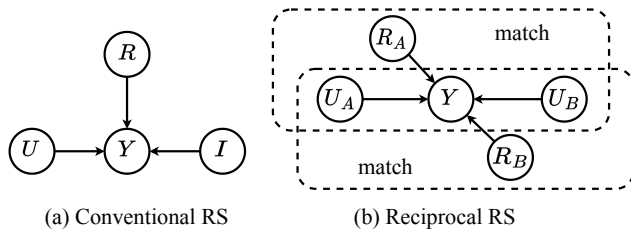
Figure 1: Causal graph for (a) recommendation as treatment in convention RS; (b) reciprocal recommendation as bilateral treatments in RRS. U: user, I: item, R: recommendation, R_A and R_B: recommendations made for each side, Y: ranking score (e.g., the probability of matching). For simplicity, we ignore all confounding factors.

Distinct from conventional user-item recommender systems [16, 27], RRS typically involves two parties of users and aims to generate mutually beneficial recommendations. Each party within the system exhibits unique preferences, requirements, and responses to the recommendations provided. Despite different task settings, most studies on RRS typically follow existing evaluation paradigms [50] (especially similar evaluation metrics), except that the evaluation process is conducted for each party independently.

Despite the simplicity and popularity, we argue that such an evaluation approach is inherently problematic since the recommendation results of both parties are coupled while the single-sided evaluation process is independent. Take online recruitment as an example and assume that a candidate sees the recruitment ad and the recruiter interviews and accepts him. When a match already occurs between the candidate and recruiter incurred by a recommendation to the candidate, there is no need to make another recommendation for the recruiter, which actually becomes redundant. However, in such a case, independent evaluations would consider both-sided recommendations as two successful recommendations, even though it doesn't increase the total number of matches made by the system. In the reciprocal scenario, an ideal goal should be the total number of matched pairs [33]. Thus, a more comprehensive evaluation perspective taking into account the overall number of matches and the degree of recommendation redundancy is needed for accurately measuring RRS performance.

In this work, we aim to evaluate the effectiveness of RRS models from a holistic perspective. We propose to consider three additional aspects for evaluation: the coverage of the overall matching relationship, the duplication degree of the recommendations, and the ranking performance with population imbalance of the two sides, which we define as *overall coverage*, *bilateral stability*, and *balanced ranking*, respectively. The underlying purpose of the overall coverage and bilateral stability metrics is to identify and filter *redundant recommendations*, *i.e.,* pairs of users that are recommended to each other on both sides. The balanced ranking aims to evaluate the performance of ranking lists while ensuring equal treatment of users on both sides, thereby mitigating bilateral imbalances.

In addition to evaluating RRS from a holistic perspective, it is essential to study how recommendations in this scenario will affect the overall effectiveness when designing RRS models. Compared

with conventional recommender systems, it becomes more intricate to model the effects of potential recommendations in RRSs. For example, when a matching relationship occurs with bilateral recommendations, it can be attributed to the recommendations made on either side or both sides. Without deeply understanding the underlying mechanism, it isn't easy to accurately model the two-sided matching process. To better understand the differences between conventional and reciprocal recommender systems, we illustrate RRS from a causal perspective [21] in Figure 1. Compared to the conventional setting, the recommendations in RRS for both sides may collaboratively influence the matching outcome, offering increased likelihood for a match. This combined effect of bilateral recommendations can be viewed as an effect of mixed treatments on the outcomes, which would affect the final results in an integrated manner. It motivates us to consider such a causal effect of bilateral recommendations for improving RRS.

To address the above issues, we propose a model-agnostic learning approach, namely **C**ausal **R**eciprocal **R**ecommender **S**ystem (**CRRS**), formulated under the *potential outcome framework* [28]. The core idea is to estimate the matching probability under different treatment assignments, which are termed the potential outcomes. There are two major technical contributions in our approach. Firstly, we propose an adaptable framework for fine-grained modeling of causal effects. This approach can be extended to existing typical recommendation system models, capturing the impact of different recommendation strategies on the outcomes. Secondly, we design an effective reranking strategy for refining the ranking score for both sides. This strategy comprehensively considers the recommendation mechanism within the reciprocal scenario, aiming to enhance the overall performance of the RRS.

The contributions of this paper can be summarized as follows:

• We introduce a comprehensive set of metrics designed to evaluate RRS holistically, addressing aspects traditionally neglected by standard metrics in existing recommender systems.

• We take a causal perspective to model the matching effect in RRS and develop a model-agnostic framework, enabling improved prediction and user ranking by viewing reciprocal recommendations as bilateral treatments.

• We conduct extensive experiments to demonstrate the effectiveness of our proposed metrics and methods compared with reciprocal recommendation models on two real-world datasets.

2 Related Work

Reciprocal Recommender System. Reciprocal Recommender Systems (RRS) have been widely studied for two-sided markets [17, 20, 32, 47], recommending users to other users rather than items. RRS are frequently employed in domains where both sides of the market have their preferences and interactions, such as online dating [14, 23, 24, 35, 41], recruitment [4, 11, 17, 33, 45, 47] and social networks [1, 9]. Since the reciprocal recommendation task can be regarded as two separate tasks, most RRS evaluate their methods individually for each task [13, 19, 24, 45, 50, 51], with widely used top-*K* recommendation metrics such as Recall, Precision, and Normalized Discounted Cumulative Gain (NDCG). However, these two tasks are two sides of a coin and the final result is determined by both sides of the market. For a better evaluation of the whole

system, studies have been conducted that build upon the economic and social science literature [6, 18, 33, 34], referring to this scenario as a *matching market*. However, these methods heavily rely on the ground truth of relevance, often unavailable in real-world datasets. In this work, we propose to evaluate RRS methods from an overall perspective, addressing overall performance across three aspects: overall coverage, bilateral stability, and balanced ranking.

Causal Inference in Recommendation. Recently, causal inference [22] has gained significant attention in recommender systems research [2, 15, 37]. The current works have explored the application of causal inference to address various challenges in recommender systems, including bias [3, 30, 53], fairness [12, 38], and explainability [40, 44]. Generally, causal inference encompasses two primary directions: causal discovery and causal estimation [46]. Causal discovery aims to learn causal relationships from data, identifying causal graphs that reveal interdependencies among factors [8, 40, 43]. Causal estimation seeks to estimate the treatment effect, especially evaluating how interventions or treatments influence user outcomes in recommender systems [7, 29, 31, 42]. Inspired by advances in causal effect estimation, our work concentrates on applying this approach to the domain of reciprocal recommendation. In this paper, we attempt to estimate the causal effects from both sides of RRS to enhance recommendation quality.

3 Problem Definition

In reciprocal recommender systems (RRS), there are two involved parties (typically two sets of users) with mutual selection relationship, denoted as $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ and $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$, where n and m denote the numbers of candidates on each side. The two parties are associated with a set of directed user-to-user interactions represented by $\mathcal{I} = \{\langle a_i, b_j, d_{ij}, r_{ij} \rangle\}$. Each tuple represents an interaction between a user $a_i \in \mathcal{A}$ and a user $b_j \in \mathcal{B}$, where the binary variable $d_{ij} \in \{0, 1\}$ indicates the direction of the interaction (1 for $a_i \rightarrow b_j$ and 0 for $b_j \rightarrow a_i$) and $r_{ij} \in \{0, 1\}$ represents whether a final match is achieved.

In the scenario of reciprocal recommendation, we perform personalized recommendations for users of both sides. Specifically, for a given user a_i, the system generates a ranked list of recommendations from $\mathcal{B}$. The other side is similar, where users from $\mathcal{A}$ are ranked when recommending to user b_j. Most existing works follow conventional evaluation paradigms to assess the ranking performance of each side individually. However, in a reciprocal scenario, the primary goal should be the overall count of matching pairs. This work focuses on evaluating the recommendations from both sides from an overall system perspective. This includes how many actual match pairs are covered by the recommendations (coverage) and whether they are recommended to each other (stability).

4 Revisiting the Evaluation Metrics

As discussed in Section 1, to evaluate the performance of RRS, existing methods mainly focus on either single-sided metrics or solely the overall quantity of matches, which lack consideration of more robust two-sided measurements, such as coverage and stability. To ensure a more holistic and robust evaluation, we propose to consider three new evaluation aspects, namely overall coverage, bilateral stability, and balanced ranking.

4.1 Proposed Metrics

Measures of Overall Coverage. Overall coverage refers to the extent to which a recommender system covers potential matching relationships from an overall perspective. To achieve a more accurate evaluation of the coverage in RRS, the successful matching led by the bilateral recommendation, where the two users involved are recommended to each other simultaneously, should be treated as a single successful recommendation, rather than being counted twice. Considering this, we propose two metrics, namely *Coverage-adjusted Recall* (CRecall) and *Coverage-adjusted Precision* (CPrecision). The corresponding formulas are given as follows:

$$\text{CRecall} = \frac{\text{TP}_A + \text{TP}_B - \text{TP}_{A \cap B}}{\text{M}}, \tag{1}$$

$$\text{CPrecision} = \frac{\text{TP}_A + \text{TP}_B - \text{TP}_{A \cap B}}{(n + m)K}, \tag{2}$$

where M denotes the total count of the matching pairs and K denotes the length of the recommendation list. $\text{TP}_A = \sum_{i \in \mathcal{A}} \text{TP}_i$ and $\text{TP}_B = \sum_{j \in \mathcal{B}} \text{TP}_j$ denote the sum of true positives for side A and side B, where TP_i and TP_j are the true positives of the single user of the two sides. $\text{TP}_{A \cap B}$ denotes the true positives where both sides have recommendations simultaneously. As we can see, CRecall and CPrecision refine the positive pairs generated by the RRS by avoiding duplicate counting of redundant successful recommendations.

Measures of Bilateral Stability. Bilateral stability refers to the extent to which a recommender system simultaneously recommends a pair of users to each other. Stability is mainly related to the number of mutual recommendations, which potentially increases the likelihood of successful matching for the involved pairs. To effectively evaluate the bilateral stability of RRS, we propose two metrics, namely *Stability-adjusted Recall* (SRecall) and *Stability-adjusted Precision* (SPrecision). The formulas are given as:

$$\text{SRecall} = \frac{\text{TP}_{A \cap B}}{\text{M}}, \tag{3}$$

$$\text{SPrecision} = \frac{\text{TP}_{A \cap B}}{(n + m)K}. \tag{4}$$

SRecall and SPrecision directly show how well the system simultaneously offers recommendations for the two sides of the given pair, effectively capturing the essence of bilateral stability.

Measures of Balanced Ranking. Balanced ranking refers to the ranking performance of the recommendation list ensuring equity across different group sizes. An essential characteristic of the RRS is the bilateral imbalances and crowding [33]. For instance, in a recruitment scenario, there might be 1000 candidates and merely 100 jobs, resulting in a significant disparity in the number of users. To fairly evaluate the ranking performance of each user in the RRS, we propose *Reciprocal* NDCG (RNDCG) based on the widely-used ranking metric NDCG [39]. It offers a more comprehensive evaluation of overall ranking performance, considering weight disparities between different sides. The formula is presented as:

$$\text{RNDCG@K} = \frac{n \cdot \text{NDCG}_A + m \cdot \text{NDCG}_B}{n + m}. \tag{5}$$

RNDCG is a weighted sum of NDCG according to the number of users on each side. Compared with NDCG, RNDCG enhances
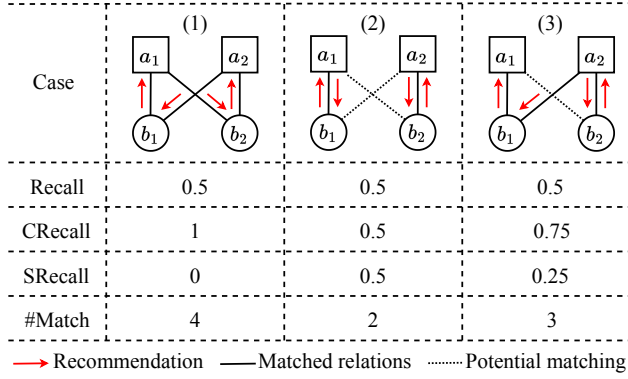
Figure 2: Three distinct recommendation cases with identical matching relationships. In this scenario, we make a top-1 recommendation for four users (a_1, a_2, b_1, and b_2) and then report the average Recall, CRecall, SRecall, and the count of successful matching pairs.

fairness and effectiveness in the recommendations, making the system more responsive to the actual distribution of users and their potential impact on the market dynamics.

Summary. In the above, we have presented the metrics of *overall coverage* (CRecall/CPrecision), *bilateral stability* (SRecall/SPrecision), and *balanced ranking* (RNDCG). Among these proposed metrics, CRecall/CPrecision along with SRecall/SPrecision aim to enhance the evaluation of RRS, aiming to promote more balanced and mutually beneficial RRS models. Furthermore, RNDCG allows us to assess the ranking performance of unbalanced groups while accounting for the crowding, ultimately leading to more informed and equitable recommendations for all users.

4.2 Effectiveness of the Proposed Metrics

To better understand the advantage of the proposed metrics, we construct three illustrative scenarios as depicted in Figure 2. For clarity, we consider a top-1 recommendation scenario, where each user receives just one recommendation (indicated by arrows). Here, matches cannot be realized without recommendations (namely potential matching, indicated by black dashed lines).

The three scenarios represent distinct recommendation strategies. In case 1, four separate recommendations each successfully established a matched relation. In case 2, two mutual recommendations resulted in two matched relations while in case 3, three matched relations were successfully established. We calculate both traditional and proposed metrics for these cases. For simplicity, we take Recall, CRecall, and SRecall for comparisons, and the other metrics can be conducted similarly. It is evident that traditional metrics fail to capture the differences among the three cases (all having Recall=0.5), whereas our proposed metrics CRecall and SRecall can capture such a distinction (*e.g.,* CRecall=1 in case 1 and CRecall=0.5 in case 2). Thus, these proposed metrics can provide a more comprehensive evaluation of the RRS, surpassing the limitations of traditional metrics.

In traditional RRS, when a pair is recommended by both sides and achieves a match, it is unclear through which mechanism the match

was realized. This uncertainty introduces challenges in understanding the effectiveness of the recommendation process, making it difficult to enhance and refine the system's performance.

5 Proposed Method

In the above, we have introduced several new evaluation metrics for improving the evaluation of RRS. Next, we study how to develop more effective reciprocal recommendation models, so as to achieve better performance on RRS. Compared with conventional recommendation tasks, reciprocal recommendation becomes more intricate in the matching mechanism, due to the potential effects of two-sided recommendation. In order to achieve a better recommendation performance, it is key to effectively model and analyze the effects of the potential factors in deriving a matching relationship. For this purpose, we formulate the reciprocal recommendation scenario from a causal perspective and then introduce the proposed **C**ausal **R**eciprocal **R**ecommender **S**ystem (named **CRRS**) under the potential outcome framework [28]. The overall architecture of the proposed approach is shown in Figure 3.

5.1 Recommendations as Bilateral Treatments

As previously mentioned, the primary goal of an RRS is to optimize the coverage of successful matching relationships. This goal is highly affected by the recommendation process, either single-sided or two-sided recommendations. Given a successful matching with bilateral recommendations, it is intricate to infer which recommendation actually leads to the matching. To better capture the underlying effects of potential factors, we view the recommendations as treatments from a causal perspective [30, 36]. In this scenario, treatments consist of two parts: recommendations for side A (denoted as T_A) and recommendations for side B (T_B). Formally, we define *bilateral treatment* as follows:

Definition 5.1 (Bilateral Treatments). For a given user pair (a_i, b_j), T_A represents whether the system recommends user b_j to a_i, while T_B represents the recommendation for the other side, *i.e.,*

$$T_A = 1 \leftrightarrow \text{recommending } b_j \text{ to } a_i, \tag{6}$$

$$T_B = 1 \leftrightarrow \text{recommending } a_i \text{ to } b_j. \tag{7}$$

The coupling of these two distinct treatments influences the outcomes profoundly and results in four unique instances of intervention (*i.e.,* $T_A T_B \in \{10, 11, 01, 00\}$). Based on the causal formulation, we next estimate the causal effect of each treatment combination and calculate the ranking scores for users.

5.2 Causal Reciprocal Recommender System

To decouple these treatments and model them separately at a finer granularity, we leverage the *potential outcome framework* [28] to analyze and estimate the causal effect. The core idea is to estimate the causal effect of recommendation in a collaborative way, under the assumption that similar users have similar treatment effects under recommendations [42]. By developing a collaborative causal solution, our approach is model-agnostic, ensuring adaptability and versatility. After the causal effect is estimated, we need to determine the final ranking list presented to both sides' users. To improve the overall effectiveness of the system, we further propose a reranking
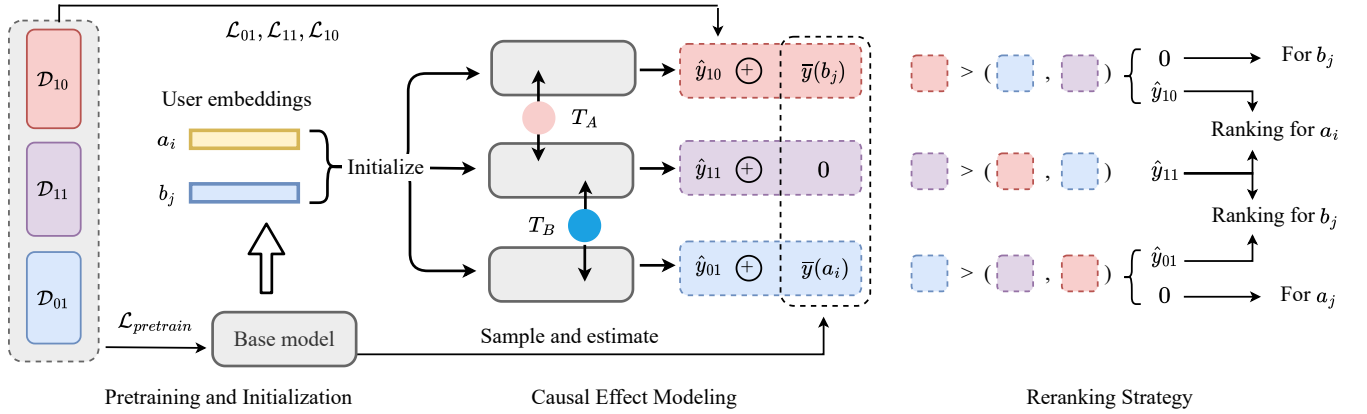
Figure 3: An illustration of the proposed framework CRRS. We represent three treatment assignments with red, purple, and blue. Each assignment signifies a distinct dataset and outcome obtained under the treatment conditions.

Table 1: An illustration of the expectation of matching with the bilateral treatments. Each unit comprises two parts: the expected outcome of the intervention and the expected outcome of other recommendations in the vacant slots.

Treatment		Outcome	Matching Expectation with Vacant Slots	
T_A	T_B	$\hat{y}$	A side	B side
1	0	$\hat{y}_{10}$	0	$\bar{y}(b_j)$
1	1	$\hat{y}_{11}$	0	0
0	1	$\hat{y}_{01}$	$\bar{y}(a_i)$	0
0	0	$\hat{y}_{00}$	$\bar{y}(a_i)$	$\bar{y}(b_j)$

strategy aimed at enhancing overall performance and choosing the best recommendation strategy.

5.2.1 Potential Outcome Framework. In causal inference, potential outcome framework [28] is a method that directly calculates the effect of certain treatments for outcomes. In recommendation scenarios, *outcome* is often defined as either implicit or explicit user feedback, such as click or rating [43]. When considering a reciprocal recommendation scenario, we define outcomes as matching relationships, as this is the most effective feedback observed. Formally, we give the definition of *potential outcome* in RRS as follows:

Definition 5.2 (Potential Outcome for RRS). Given the treatment assignment $T = (T_A, T_B)$, the potential outcome Y_T denotes matching relationship under the specific recommendation strategy where $T = t \in \{10, 11, 01, 00\}$ for instance user pair (a_i, b_j). The potential outcomes can be instantiated as a function $Y_T = f_T(x_i, x_j)$, which takes user features x_i and x_j as input, *i.e.*, $\hat{y}_t = f_t(x_i, x_j)$.

We present the correspondence of $\hat{y}$ to treatment t in Table 1. Specifically, an estimation of $\hat{y}_{10} = 1$ indicates that a match is expected when the treatment combination is assigned as $T_A = 1$ and $T_B = 0$. Conversely, the estimation of $\hat{y}_{00}$ consistently remains at 0 because no match is expected when neither side receives a recommendation. To model these bilateral treatments, a major challenge is that there is only one intervention per sample in RRS. We cannot observe the final results of different treatments for the same pair

of users at the same time, which is a counterfactual problem [43]. Therefore, it is challenging to estimate counterfactual outcomes and calculate the matching probability for reciprocal recommendations. However, based on the intuition that similar users correspond to similar treatment effects [42], we can estimate the causal effect of three different treatment assignments with typical RS models.

5.2.2 Causal Effect Modeling and ranking. As discussed before, we reformulate the RRS scenario with the potential outcome framework and regard bilateral recommendations as treatments. Under the four treatment assignments mentioned above, the cases that we have to consider that are likely to be successful are those with $t \in \{10, 11, 01\}$. As shown in Figure 1, $t = 10$ is to recommend b_j to a_i, $t = 01$ is to recommend a_i to b_j, and $t = 11$ is to recommend both users to each other. When $t = 00$, the pair cannot be matched because neither side is recommended to the other. To support this fine-grained modeling and predict the potential outcomes under the three treatments, we construct three functions:

$$\hat{y}_{10} = f_{10}(a_i, b_j), \; \hat{y}_{11} = f_{11}(a_i, b_j), \; \hat{y}_{01} = f_{01}(a_i, b_j), \quad (8)$$

where each $f_t(\cdot)$ represents a specific recommender model backbone and each predicted value ranges from 0 to 1. The three backbone models essentially have few restrictions and can be implemented by typical recommendation models (*e.g.*, BPRMF [25] or LightGCN [10]). Nonetheless, they possess unique parameters, which allow them to outperform a single model in effectively predicting outcomes for different treatments. Our proposed method is model-agnostic and can be naturally applied to typical recommendation models. Following this paradigm, the expectations of matching under different recommendation strategies can be estimated.

After the three outcomes have been predicted, we need to rank the candidates for each user on both sides. For a given user pair (a_i, b_j), the scores for ranking can be simply calculated using the following formula:

$$s_{a_i} = \hat{y}_{10} + \hat{y}_{11}, \; s_{b_j} = \hat{y}_{01} + \hat{y}_{11}, \quad (9)$$

where s_{a_i} and s_{b_j} are the scores of recommendations for a_i and b_j, respectively. Each score is the aggregate of two scores calculated

under the treatment for which the recommendation is provided, either as a direct sum or alternatively, as a weighted sum.

5.2.3 Reranking Strategy with Vacant Slots.
Although the aforementioned approach facilitates ranking, it still falls short of fully addressing the recommendation mechanism of this scenario.

In practice, if one side of a pair (e.g., a_i and b_j) lacks recommendations (e.g., $T_A = 0$), there will be an empty spot for the recommendation and we term this a *vacant slot*. Another user (say b_v) will be recommended to a_i at this vacant slot. Such a recommendation of b_v introduces a new possibility of matching denoted as $\bar{y}(a_i)$, which is the matching expectation of the pair (a_i, b_v). Consequently, the global expectation of matching becomes $\hat{y}_{01} + \bar{y}(a_i)$. The results under other treatments are shown in Table 1. With the potential outcome predictions and matching expectations brought by the vacant slots, we need to choose and refine the scores for ranking users for the two sides. The formula for ranking scores for the two sides is set as follows:

$$s_{a_i} = \text{IsMax}(\hat{y}_{11}) \cdot \hat{y}_{11} + \text{IsMax}(\hat{y}_{10} + \bar{y}(b_j)) \cdot \hat{y}_{10}, \quad (10)$$

$$s_{b_j} = \text{IsMax}(\hat{y}_{11}) \cdot \hat{y}_{11} + \text{IsMax}(\hat{y}_{01} + \bar{y}(a_i)) \cdot \hat{y}_{01}, \quad (11)$$

where $\text{IsMax}(\cdot)$ is a boolean value for the set $\mathcal{S} = \{\hat{y}_{11}, \hat{y}_{01} + \bar{y}(a_i), \hat{y}_{10} + \bar{y}(b_j), \bar{y}(a_i) + \bar{y}(b_j)\}$, denoting the best recommendation strategy for a given pair:

$$\text{IsMax}(y) = \begin{cases} 1 & \text{if } y = \max(\mathcal{S}), \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The aim of IsMax is to ensure that the final recommendation method is targeted at maximizing overall matching performance. To estimate $\bar{y}(a_i)$ and $\bar{y}(b_j)$, we sample a certain number of users from the opposite side and then calculate their relevance scores using the backbone model. This estimation serves as an approximation of the $\bar{y}(a_i)$ and $\bar{y}(b_j)$ based on the sampled users. By considering both relevance and direction, the reranking process ensures rankings with the most matching expectations, taking into consideration the reciprocal nature of the RRS.

5.3 Optimization

Our proposed CRRS approach consists of two learning stages: pre-training and counterfactual learning. In the pre-training stage, we use real matching labels to train a backbone recommender model. In the counterfactual learning stage, we finetune the recommender model with direction labels to better predict outcomes, taking into account the causal effects of various recommendation strategies. The full algorithm is described in Algorithm 1.

5.3.1 First Stage: Pre-training.
We optimize the backbone model using the Bayesian Personalized Ranking (BPR) loss [25], which effectively captures the relative preferences between positive and negative samples. To describe the pre-training process, we take side A for illustration when it is symmetric for side B, we define the pre-training loss $\mathcal{L}_{pretrain}$ as follows:

$$\mathcal{L}_{pretrain} = -\frac{1}{|\mathcal{D}|} \sum_{(a_i, b_j) \in \mathcal{D}} \log\left(\sigma(\hat{y}(a_i, b_j) - \hat{y}(a_i, b_{j'}))\right), \quad (13)$$

Algorithm 1: The learning algorithm of **CRRS**.

---

input: User set $\mathcal{A}$ and User set $\mathcal{B}$, interaction set $\mathcal{D}_{11}$ with bilateral treatment, $\mathcal{D}_{01}$ with treatment for A side and $\mathcal{D}_{10}$ with treatment for B side.

1 Pre-train the backbone model with relevance r_{ij}.
2 Initialize three models under different treatment assignments with the same pre-trained user embeddings a_i and b_j. # (Sec. 5.3.1)
3 **foreach** *mini-batch* $\mathcal{M}$ **do**
4 $\quad \mathcal{L}_t \leftarrow 0, t \in \{11, 10, 01\}$
5 $\quad$ Calculate $\mathcal{L}_{pretrain}$ with $\mathcal{M}$.
6 $\quad$ **foreach** *user-user pair* $(a_i, b_j) \in \mathcal{M}$ **do**
7 $\quad\quad$ Random get a negative sample $b_{j'}$ for A side and a negative sample $a_{i'}$ for B side.
8 $\quad\quad$ **if** $(a_i, b_j) \in \mathcal{D}_{01}$ **then**
9 $\quad\quad\quad$ $\mathcal{L}_{01} \leftarrow \mathcal{L}_{01} - \sigma(\hat{y}_{01}(a_i, b_j) - \hat{y}_{01}(a_i, b_{j'}))$
10 $\quad\quad$ **if** $(a_i, b_j) \in \mathcal{D}_{10}$ **then**
11 $\quad\quad\quad$ $\mathcal{L}_{10} \leftarrow \mathcal{L}_{10} - \sigma(\hat{y}_{10}(a_i, b_j) - \hat{y}_{10}(a_{i'}, b_j))$
12 $\quad\quad$ **if** $(a_i, b_j) \in \mathcal{D}_{11}$ **then**
13 $\quad\quad\quad$ $\mathcal{L}_{11} \leftarrow \mathcal{L}_{11} - \sigma(\hat{y}_{11}(a_i, b_j) - \hat{y}_{11}(a_i, b_{j'}))$
14 $\quad\quad\quad$ $\mathcal{L}_{11} \leftarrow \mathcal{L}_{11} - \sigma(\hat{y}_{11}(a_i, b_j) - \hat{y}_{11}(a_{i'}, b_j))$
15 $\quad\quad\quad$ $\mathcal{L}_{01} \leftarrow \mathcal{L}_{01} - \sigma(\hat{y}_{01}(a_i, b_j) - \hat{y}_{01}(a_i, b_{j'}))$
16 $\quad\quad\quad$ $\mathcal{L}_{10} \leftarrow \mathcal{L}_{10} - \sigma(\hat{y}_{10}(a_i, b_j) - \hat{y}_{10}(a_{i'}, b_j))$
17 $\quad$ $\mathcal{L}_t \leftarrow \mathcal{L}_t + \mathcal{L}_{pretrain}, t \in \{11, 10, 01\}$
18 $\quad$ Backpropagate each $\mathcal{L}_t$ and update the parameters

---

where $\sigma(\cdot)$ is the sigmoid function, $\mathcal{D}$ denotes the training data, (a_i, b_j) is a positive sample for matching, and $b_{j'}$ denotes the negative sample for user a_i. This stage aims to learn a basic recommender model for subsequent causal finetuning.

5.3.2 Second Stage: Counterfactual Learning.
At this stage, we further finetune the recommendation model using data under different treatment assignments. Similar to the first stage, the counterfactual learning loss also employs the BPR ranking loss. For each backbone model, the loss function is expressed as follows:

$$\mathcal{L}_t = -\frac{1}{|\mathcal{D}_t|} \sum_{(a_i, b_j) \in \mathcal{D}_t} \log\left(\sigma(\hat{y}_t(a_i, b_j) - \hat{y}_t(a_i, b_{j'}))\right), \quad (14)$$

where $\mathcal{D}_t$ represents the training data specific to each treatment. To ensure training stability, we combine the counterfactual learning loss with the pre-training loss. The treatment-specific data provides important guidance that instructs the models to refine their performance under specific conditions. In this way, the models can more effectively discern the causal effects between treatments and outcomes, thus better achieving better performance for RRS.

5.4 Discussion

We conduct a discussion here to clarify our stance on redundant recommendations. In practice, different scenarios may exhibit varying stances toward redundant recommendations. For example, on LinkedIn[1], mutual recommendations between matched candidates and recruiters are optional. In contrast, on Tinder[2], two users can

---

[1]https://linkedin.com/
[2]https://tinder.com/

Table 2: Statistics of the experimental datasets.

Dataset	#User A	#User B	#Interaction	#Match	Sparsity
Recruitment	32, 161	25, 665	790, 725	224, 636	99.90%
Dating	6, 391	6, 516	605, 288	51, 474	98.55%

only start communicating if they both recommend each other and click the "like" button. In this paper, we adopt a negative stance towards redundant recommendations, prioritizing the enhancement of overall coverage and achieving higher quantities as fundamental objectives (measured by CRecall and CPrecision), despite potential trade-offs in stability (SRecall and SPrecision), as evidenced in subsequent experiments.

6 Experiments

In this section, we aim to evaluate the proposed CRRS's performance and the proposed metrics' effectiveness. We start with the experiment setup, followed by a thorough evaluation comparing the CRRS with baselines. Then we analyze traditional versus proposed metrics and present results from ablation studies alongside a ranking analysis of redundant recommendations.

6.1 Experimental Setup

Dataset. To evaluate the effectiveness of our method, we conduct experiments on two large real-world datasets from different reciprocal recommendation domains: recruitment and online dating.

• **Recruitment** [45]: This dataset contains two weeks of real logs collected from a large and popular online recruitment platform, which records dynamic interactions between job-seeking candidates and proactive recruiters.

• **Dating**[3]: This dataset is collected from the popular online dating service Libimseti. It encompasses ratings between users, and we only keep ratings between opposite-sex pairs (*i.e.,* female-to-male and male-to-female). Pairs with mutual ratings of 8 or above are viewed as a successful match, given the increased likelihood of a match with bilateral exposure.

Following the standard practice [26, 52], we only keep the 5-core interactions for both datasets. After preprocessing, we randomly split both datasets into training, validation, and test sets with ratios of 8 : 1 : 1. The statistics of both datasets are summarized in Table 2.

Baseline models. As our proposed CRRS is model-agnostic, we can apply it to various recommendation models. We choose the following two popular models as our backbone models:

• **BPRMF** [25]: This model designs Bayesian Personalized Ranking (BPR) loss to optimize the Matrix Factorization (MF), which is representative in the collaborative filtering recommendation.

• **LightGCN** [10]: This model designs a simple but competitive GCN architecture for the graph-based recommendation.

We also compare our proposed CRRS with the following reciprocal recommendation models:

• **D-BPRMF**: This method leverages two same BPRMF but with different parameters to make recommendations for each side.

[3]http://konect.cc/networks/libimseti/

• **LFRR** [19]: LFRR is a latent factor model designed specifically for reciprocal recommendation scenarios, which aggregates two MFs to model the preferences of both sides involved.

• **D-LightGCN**: This method is similar to D-BPRMF, but it utilizes LightGCN as the backbone model.

• **DPGNN** [45]: DPGNN is a graph-based method to model the two-way selection preference through a dual-perspective graph representation learning approach.

Note that these baseline models fall into two categories: MF-based Models and graph-based Models, where the backbone models used are BPRMF and LightGCN, respectively. The models with the "D-" prefix employ two identical backbone models to train and make predictions independently for each side. LFRR and DPGNN can be viewed as extensions of BPRMF and LightGCN to reciprocal recommendation scenarios, respectively. To be more inclusive, we just opted for models in the general setting rather than those designed specifically for sequences [4, 50], which have more complex data and model requirements.

Evaluation settings. To evaluate the performance of the top-K recommendation of RRS, we employ both traditional metrics Recall, Precision, NDCG, and metrics we proposed in Section 4. For each dataset, we employ a full-ranking evaluation where every candidate user is ranked based on its relevance for the target user. We first report on single-sided metrics, treating RRS as two independent ranking tasks: ranking all users from side B for each user $a_i \in \mathcal{A}$ and vice versa. Meanwhile, we record the average values of the single-sided metrics. Following this, we record the five proposed metrics: CRecall, CPrecision, SRecall, SPrecision, and RNDCG, to measure the comprehensive capability of the models. Additionally, we report the number of matching pairs under recommendations (denoted as True Positive Pairs).

Implementation details. We implement baseline models using a popular open-source recommendation library RECBOLE [48, 49]. For a fair comparison, the dimension of user embedding is standardized to 128 across all models. K is set to 50 empirically. Moreover, we optimize all methods using the Adam optimizer and conduct a thorough search for the hyper-parameters of all baselines. The learning rate is tuned from {0.001, 0.0001, 0.00001} for optimal performance. To prevent overfitting, we employ early stopping with patience of 30 epochs. To facilitate evaluations in offline experiments, we consider that all positive instances can be successfully matched when recommendations are made exclusively on one side during both the training and evaluation phases.

6.2 Experimental Results

6.2.1 Overall Performance. We conduct a comprehensive comparison of different methods across various evaluation metrics. The results on two datasets are reported in Table 3.

Performance comparison between models. As we can see, in *Recruitment*, reciprocal methods LFRR and DPGNN demonstrate superior performance across most single-sided evaluation metrics, due to their ability to effectively capture the underlying interactions. Our proposed CRRS is not necessarily better than a well-designed reciprocal recommendation model in single-sided metrics but achieves the highest performance in overall coverage metrics

Table 3: Performance comparison of all methods on Recruitment and Dating dataset. The best performance and the second best performance methods are denoted in bold and underlined fonts, respectively. "∗" indicates that the improvements are statistically significant with paired t-test (p-value < 0.05). "True Positive Pairs" represent the count of matching pairs.

Dataset	Direction	Metric	MF-based Model				Graph-based Model			
			BPRMF	D-BPRMF	LFRR	CRRS (Ours)	LightGCN	D-LightGCN	DPGNN	CRRS (Ours)
Recruitment	A side	Recall@50	0.2424	0.1223	**0.2569**∗	0.2309	0.2688	0.1929	**0.3621**∗	0.3560
		Precision@50	0.0067	0.0035	**0.0071**	0.0059	0.0072	0.0053	0.0090	**0.0093**
		NDCG@50	0.1070	0.0632	**0.1141**∗	0.1077	0.0997	0.0741	**0.1573**∗	0.1276
	B side	Recall@50	0.1886	0.1282	0.2001	**0.2355**∗	0.1764	0.1384	**0.3038**∗	0.2757
		Precision@50	0.0063	0.0043	0.0067	**0.0072**∗	0.0056	0.0045	**0.0093**∗	0.0088
		NDCG@50	0.0887	0.0606	0.0945	**0.1016**∗	0.0625	0.0533	**0.1288**∗	0.0973
	Average	Recall@50	0.2155	0.1253	0.2285	**0.2332**∗	0.2226	0.1656	**0.3329**∗	0.3158
		Precision@50	0.0065	0.0039	**0.0069**∗	0.0066	0.0064	0.0049	**0.0091**	0.0090
		NDCG@50	0.0978	0.0619	0.1043	**0.1046**	0.0811	0.0637	**0.1431**∗	0.1124
	Overall	SRecall@50	0.0617	0.0152	**0.0967**∗	0.0242	0.0638	0.0414	**0.1535**∗	0.1248
		SPrecision@50	0.0021	0.0005	**0.0029**∗	0.0006	0.0018	0.0013	**0.0042**∗	0.0036
		RNDCG@50	0.0989	0.0620	**0.1054**	0.1050	0.0832	0.0649	**0.1447**∗	0.1142
		CRecall@50	0.3388	0.2336	0.3530	**0.3968**∗	0.3609	0.2811	0.4555	**0.4670**∗
		CPrecision@50	0.0052	0.0036	0.0054	**0.0061**∗	0.0056	0.0043	0.0070	**0.0073**∗
		True Positive Pairs	7,610	5,247	7,929	**8,913**∗	8,106	6,314	10,231	**10,490**∗
Dating	A side	Recall@50	0.1760	0.1211	0.1738	**0.1783**∗	0.1787	0.1719	0.1888	**0.2172**∗
		Precision@50	0.0073	0.0053	0.0074	**0.0077**∗	0.0070	0.0071	0.0076	**0.0088**∗
		NDCG@50	0.0756	0.0052	0.0719	**0.0809**∗	0.0769	0.0724	0.0788	**0.0922**∗
	B side	Recall@50	0.1343	0.0830	**0.1719**	0.1607	0.1233	0.1616	0.1762	**0.1916**∗
		Precision@50	0.0069	0.0047	**0.0083**	0.0082	0.0059	0.0077	0.0079	**0.0089**∗
		NDCG@50	0.0566	0.0360	**0.0693**	0.0692	0.0499	0.0676	0.0689	**0.0777**∗
	Average	Recall@50	0.1552	0.1021	**0.1729**	0.1695	0.1510	0.1668	0.1825	**0.2044**∗
		Precision@50	0.0071	0.0050	0.0079	**0.0080**	0.0065	0.0074	0.0078	**0.0089**∗
		NDCG@50	0.0661	0.0206	0.0706	**0.0751**	0.0634	0.0700	0.0739	**0.0850**∗
	Overall	SRecall@50	0.0830	0.0419	0.0968	**0.0993**∗	0.0753	0.0732	0.0997	**0.1221**∗
		SPrecision@50	0.0018	0.0009	0.0021	**0.0022**	0.0017	0.0016	0.0022	**0.0027**∗
		RNDCG@50	0.0660	0.0207	0.0706	**0.0751**∗	0.0633	0.0700	0.0738	**0.0849**∗
		CRecall@50	0.2795	0.2055	0.3045	**0.3086**∗	0.2549	0.2986	0.3007	**0.3387**∗
		CPrecision@50	0.0062	0.0045	0.0067	**0.0068**	0.0056	0.0066	0.0067	**0.0075**∗
		True Positive Pairs	1,439	1,058	1,567	**1,588**∗	1,312	1,537	1,548	**1,743**∗

CRecall and CPrecision and the count of matching pairs. For the *dating*, owing to the behavioral direction characteristics in the data and there being relatively more user interactions, the proposed method outperforms others across all metrics. This can be attributed to the continued training stage, which enables the model to better capture the correlations between users on both sides. Moreover, the D-BPRMF and D-LightGCN exhibit almost the worst performance across all metrics, largely due to their insufficient training data.

Trade-off between coverage and stability. In *Recruitment* dataset, a phenomenon emerges when comparing CRRS with the best baseline model DPGNN: there is a rise in CRecall (0.4670 *v.s.* 0.4555), contrasted by a fall in SRecall (0.1248 *v.s.* 0.1535), similar to CPrecision and SPrecision. This phenomenon is normal when trying to reduce redundant recommendations as analyzed in Section 4.2. However, this phenomenon does not occur in *dating* dataset, possibly because there is a significant difference in sparsity between the two datasets on "#Match" (99.90% *v.s.* 98.55%), with even greater sparsity differences on "#Interactions". The inherent conflict between coverage and stability exists due to the different stances towards redundant recommendations, suggesting that trade-offs may be required in certain situations.

6.2.2 Metric Comparative Analysis. We conduct experiments to compare traditional metrics and the proposed metrics under different recommendation strategies on *dating*. For the same model, we manually adjust the redundant recommendations and consider the following two adjustments: **(A)** BPRMF-Uni: Convert the redundant recommendations with another matching pair that does not appear on both sides. **(B)** BPRMF-Rep: Convert the non-redundant pairs with other matching pairs that have appeared on one side. The original model is named as BPRMF-Ori. As shown in Figure 4, We depict a rectangular interaction diagram with users from both sides represented along the horizontal and vertical axes. Blue points are true positives for just one side and the green ones are true positives for the two sides. Recommendations on three cases have completely identical traditional ranking metrics (Recall, Precision, and NDCG). However, significant differences exist in the proposed metrics, which directly influence the count of matching pairs. Thus, it is critical to take the proposed metrics into account in RRS.
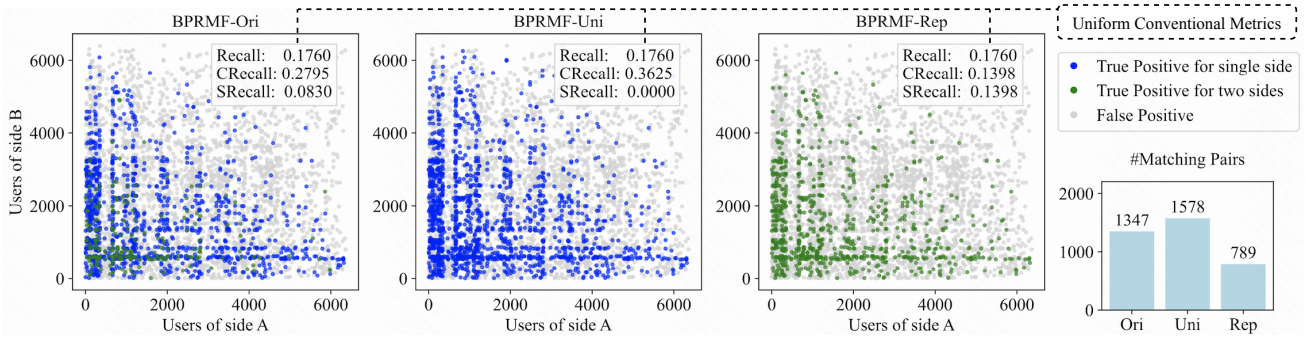
Figure 4: Comparative analysis between traditional and proposed metrics. We employ the same model BPRMF on _Dating_ and adjust the redundant recommendations by manually converting them into non-redundant ones and vice versa. In three cases, the models demonstrate identical traditional metrics yet display completely divergent overall performance.

Table 4: Effectiveness analysis of key components of CRRS. CR, CP, SR, and SP are the simplifications for CRecall, CPrecision, SRecall, and SPrecision, respectively.

Dataset	Model	Proposed Metric				
		CR@50	CP@50	SR@50	SP@50	RNDCG
Recruitment	CRRS (BPRMF)	**0.3968**	**0.0061**	0.0242	0.0006	0.1050
	w/o finetune	0.3694	0.0057	0.0397	**0.0023**	**0.1070**
	w/o pre-train	0.3400	0.0052	0.0527	0.0014	0.0689
	w/o rerank	0.3091	0.0048	**0.0757**	0.0020	0.0882
	BPRMF	0.3388	0.0052	0.0617	0.0021	0.0989
Dating	CRRS (BPRMF)	**0.3086**	**0.0068**	0.0993	0.0022	**0.0751**
	w/o finetune	0.2927	0.0065	0.0696	0.0015	0.0692
	w/o pre-train	0.2986	0.0066	0.0826	0.0018	0.0669
	w/o rerank	0.3011	0.0067	**0.1123**	**0.0025**	0.0748
	BPRMF	0.2795	0.0062	0.0830	0.0018	0.066

_6.2.3_ _Ablation Study._ The major technical contribution of our approach lies in the potential outcome framework training algorithm, as well as the proposed reranking strategy. We now analyze how each part contributes to the final performance. We consider the following three variants of CRRS: **(A)** CRRS w/o finetuning (Equation (14)); **(B)** CRRS w/o pre-training (Equation (13)); **(C)** CRRS w/o reranking (Equation (10));

In Table 4, we can observe that the absence of any component will lead to performance degradation, with varying impacts across different datasets. In _recruitment_, the reranking strategy contributes significantly to the overall coverage improvement as its objective is to avoid redundant recommendations. At the same time, we can observe that the result of the model without reranking demonstrates superior performance in bilateral stability. In _dating_, pre-training and finetuning play more vital roles in enhancing the model performance across various metrics. While the benefits brought by the reranking strategy are relatively modest. These results highlight the usefulness of all components in CRRS.

_6.2.4_ _Ranking Analysis of Redundant Recommendations._ We conduct experiments here to analyze the distribution of redundant recommendations in the ranking list. We employ the same BPRMF model to generate top-50 recommendations for both sides and record the ranking of all redundant recommendations. As discussed
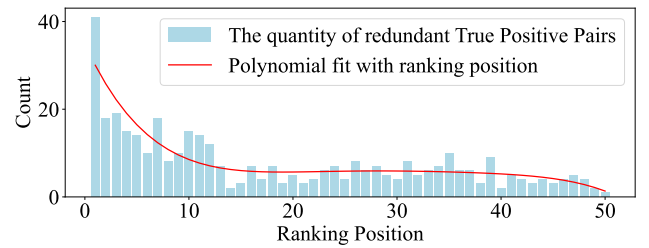

Figure 5: Distribution of ranking positions for redundant recommendations within the top 50 positions.

above, an effective way to enhance coverage involves reducing duplication between the bilateral ranking lists. However, as Figure 5 shows, the majority of redundant recommendations are found in the advanced positions, so an objective of directly optimizing the discrepancy between both sides may be ineffective. This suggests that innovative approaches like reranking are essential for achieving more effective and balanced outcomes.

7 Conclusion

This paper revisited the study of reciprocal recommender systems by introducing new metrics, formulation, and method. Firstly, we proposed five metrics for evaluating RRS from three new aspects. Then we formulated reciprocal recommendation tasks from a causal perspective, considering the recommendations as bilateral interventions. Furthermore, we proposed a causal reciprocal recommendation model using a potential outcome framework. To optimize overall performance, we further designed a reranking strategy to enhance the overall performance. Extensive experiments on two datasets indicate that the proposed approach can achieve superior performance on overall performance.

Acknowledgments

References

[1] Slah Alsaleh, Richi Nayak, Yue Xu, and Lin Chen. 2011. Improving matching process in social network using implicit and explicit user information. In *Asia-Pacific Web Conference*. Springer, 313–320.

[2] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *Proceedings of the 12th ACM conference on recommender systems*. 104–112.

[3] 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.

[4] Jiaxing Chen, Hongzhi Liu, Hongrui Guo, Yingpeng Du, Zekai Wang, Yang Song, and Zhonghai Wu. 2023. Bilateral Sequential Hypergraph Convolution Network for Reciprocal Recommendation. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 974–979.

[5] Eugene Choo and Aloysius Siow. 2006. Who marries whom and why. *Journal of political Economy* 114, 1 (2006), 175–201.

[6] David Gale and Lloyd S Shapley. 1962. College admissions and the stability of marriage. *The American Mathematical Monthly* 69, 1 (1962), 9–15.

[7] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal inference in recommender systems: A survey and future directions. *ACM Transactions on Information Systems* (2022).

[8] Jens Hainmueller. 2012. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis* 20, 1 (2012), 25–46.

[9] Jianming He and Wesley W Chu. 2010. A social network-based recommender system (SNRS). In *Data mining for social network data*.

[10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*.

[11] Xiao Hu, Yuan Cheng, Zhi Zheng, Yue Wang, Xinxin Chi, and Hengshu Zhu. 2023. BOSS: A Bilateral Occupational-Suitability-Aware Recommender System for Online Recruitment. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4146–4155.

[12] Wen Huang, Lu Zhang, and Xintao Wu. 2022. Achieving Counterfactual Fairness for Causal Bandit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 6952–6959.

[13] Kai-Huang Lai, Zhe-Rui Yang, Pei-Yuan Lai, Chang-Dong Wang, Mohsen Guizani, and Min Chen. 2024. Knowledge-Aware Explainable Reciprocal Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 8636–8644.

[14] Lei Li and Tao Li. 2012. MEET: a generalized framework for reciprocal recommender systems. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 35–44.

[15] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI. AUAI*.

[16] Jie Lu, Dianshuang Wu, Mingsong Mao, Wei Wang, and Guangquan Zhang. 2015. Recommender system application developments: a survey. *Decision support systems* 74 (2015), 12–32.

[17] Tsunenori Mine, Tomoyuki Kakuta, and Akira Ono. 2013. Reciprocal recommendation for job matching with bidirectional feedback. In *IIAI*.

[18] Martin Mladenov, Elliot Creager, Omer Ben-Porat, Kevin Swersky, Richard Zemel, and Craig Boutilier. 2020. Optimizing long-term social welfare in recommender systems: A constrained matching approach. In *International Conference on Machine Learning*. PMLR, 6987–6998.

[19] James Neve and Ivan Palomares. 2019. Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems. In *RecSys*.

[20] Iván Palomares, Carlos Porcel, Luiz Pizzato, Ido Guy, and Enrique Herrera-Viedma. 2021. Reciprocal Recommender Systems: Analysis of state-of-art literature, challenges and opportunities towards social recommendation. *Information Fusion* 69 (2021), 103–127.

[21] Judea Pearl. 2009. *Causality*. Cambridge university press.

[22] Judea Pearl. 2010. Causal inference. *Causality: objectives and assessment* (2010), 39–58.

[23] Luiz Pizzato, Tomasz Rej, Joshua Akehurst, Irena Koprinska, Kalina Yacef, and Judy Kay. 2013. Recommending people to people: the nature of reciprocal recommenders with a case study in online dating. *User Modeling and User-Adapted Interaction* 23, 5 (2013), 447–488.

[24] Luiz Pizzato, Tomek Rej, Thomas Chung, Irena Koprinska, and Judy Kay. 2010. RECON: a reciprocal recommender for online dating. In *RecSys*.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*.

[26] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[27] Paul Resnick and Hal R Varian. 1997. Recommender systems. *Commun. ACM* 40, 3 (1997), 56–58.

[28] Donald B Rubin. 1980. Randomization analysis of experimental data: The Fisher randomization test comment. *Journal of the American statistical association* 75, 371 (1980), 591–593.

[29] Masahiro Sato, Hidetaka Izumo, and Takashi Sonoda. 2016. Modeling Individual Users' Responsiveness to Maximize Recommendation Impact. In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization*. 259–267.

[30] 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.

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

[32] Zheng Siting, Hong Wenxing, Zhang Ning, and Yang Fan. 2012. Job recommender systems: a survey. In *ICCSE*.

[33] Yi Su, Magd Bayoumi, and Thorsten Joachims. 2022. Optimizing Rankings for Recommendation in Matching Markets. In *TheWebConf*.

[34] Yoji Tomita, Riku Togashi, and Daisuke Moriwaki. 2022. Matching Theory-based Recommender Systems in Online Dating. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 538–541.

[35] Kun Tu, Bruno Ribeiro, David Jensen, Don Towsley, Benyuan Liu, Hua Jiang, and Xiaodong Wang. 2014. Online dating recommendations: matching markets and learning preferences. In *WWW*.

[36] Wenjie Wang, Yang Zhang, Haoxuan Li, Peng Wu, Fuli Feng, and Xiangnan He. 2023. Causal recommendation: Progresses and future directions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3432–3435.

[37] 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.

[38] 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.

[39] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of NDCG type ranking measures. In *Conference on learning theory*. PMLR, 25–54.

[40] Zhenlei Wang, Xu Chen, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Sequential recommendation with causal behavior discovery. *arXiv preprint arXiv:2204.00216* (2022).

[41] Peng Xia, Benyuan Liu, Yizhou Sun, and Cindy Chen. 2015. Reciprocal recommendation system for online dating. In *ASONAM*.

[42] Xu Xie, Zhaoyang Liu, Shiwen Wu, Fei Sun, Cihang Liu, Jiawei Chen, Jinyang Gao, Bin Cui, and Bolin Ding. 2021. CausCF: Causal Collaborative Filtering for Recommendation Effect Estimation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4253–4263.

[43] Shuyuan Xu, Jianchao Ji, Yunqi Li, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Causal Inference for Recommendation: Foundations, Methods and Applications. *arXiv preprint arXiv:2301.04016* (2023).

[44] Shuyuan Xu, Yunqi Li, Shuchang Liu, Zuohui Fu, Yingqiang Ge, Xu Chen, and Yongfeng Zhang. 2021. Learning causal explanations for recommendation. In *The 1st International Workshop on Causality in Search and Recommendation*.

[45] Chen Yang, Yupeng Hou, Yang Song, Tao Zhang, Ji-Rong Wen, and Wayne Xin Zhao. 2022. Modeling Two-Way Selection Preference for Person-Job Fit. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 102–112.

[46] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. 2021. A survey on causal inference. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 15, 5 (2021), 1–46.

[47] Hongtao Yu, Chaoran Liu, and Fuzhi Zhang. 2011. Reciprocal recommendation algorithm for the field of recruitment. *JICS* (2011).

[48] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. *arXiv preprint arXiv:2206.07351* (2022).

[49] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*.

[50] Bowen Zheng, Yupeng Hou, Wayne Xin Zhao, Yang Song, and Hengshu Zhu. 2023. Reciprocal Sequential Recommendation. *arXiv preprint arXiv:2306.14712* (2023).

[51] Zhi Zheng, Xiao Hu, Zhaopeng Qiu, Yuan Cheng, Shanshan Gao, Yang Song, Hengshu Zhu, and Hui Xiong. 2024. Bilateral Multi-Behavior Modeling for Reciprocal Recommendation in Online Recruitment. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[52] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM*.

[53] Yaochen Zhu, Jing Ma, and Jundong Li. 2023. Causal Inference in Recommender Systems: A Survey of Strategies for Bias Mitigation, Explanation, and Generalization. *arXiv preprint arXiv:2301.00910* (2023).