Enhancing Recommendation with Search Data in a Causal Learning Manner

ZIHUA SI, ZHONGXIANG SUN, XIAO ZHANG, and JUN XU, Gaoling School of Artificial Intelligence, Renmin University of China
YANG SONG and XIAOXUE ZANG, Kuaishou Technology Co., Ltd.
JI-RONG WEN, Gaoling School of Artificial Intelligence, Renmin University of China

Recommender systems are currently widely used in various applications helping people filter information. Existing models always embed the rich information for recommendation, such as items, users, and contexts in real-value vectors, and make predictions based on these vectors. In the view of causal inference, the associations between representation vectors and user feedback are inevitably a mixture of the causal part that describes why a user prefers an item, and the non-causal part that merely reflects the statistical dependencies, for example, the display ranking position and sales promotion. However, most recommender systems assume the user-item interactions are only affected by user preferences, neglecting the striking differences between these two associations. To address this problem, we propose a model-agnostic causal learning framework called IV4Rec+ that can effectively decompose the embedding vectors into these two parts. Moreover, two strategies are proposed to utilize search queries as instrumental variables: IV4Rec+(I) only decomposes the item embeddings, while IV4Rec+(UI) decomposes both user and item embeddings. IV4Rec+ is a model-agnostic design that can be applied to many existing recommender systems, e.g., DIN, NRHUB, and SRGNN. Extensive experiments on three datasets show that IV4Rec+ significantly facilitates the performance of recommender systems and outperforms state-of-the-art frameworks.

CCS Concepts: • **Information systems → Recommender systems;**

Additional Key Words and Phrases: Recommendation, search, causal learning, instrumental variables

ACM Reference format:
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 Trans. Inf. Syst.* 41, 4, Article 111 (April 2023), 31 pages.
https://doi.org/10.1145/3582425

This work was funded by the National Key R&D Program of China (2019YFE0198200), Kuaishou, the National Natural Science Foundation of China (61872338, 62006234, 61832017), Beijing Outstanding Young Scientist Program NO. BJJWZYJH012019100020098, Intelligent Social Governance Interdisciplinary Platform, Major Innovation & Planning Interdisciplinary Platform for the "Double-First Class" Initiative, Renmin University of China, and Public Policy and Decision-making Research Lab of Renmin University of China.
Authors' addresses: Z. Si, Z. Sun, X. Zhang, J. Xu (corresponding author), and J.-R. Wen, Gaoling School of Artificial Intelligence, Renmin University of China, No. 59 Zhongguancun Street, Haidian District, Beijing, 100872, China; emails: zihua_si@ruc.edu.cn, sunzhongxiang@ruc.edu.cn, zhangx89@ruc.edu.cn, junxu@ruc.edu.cn, jrwen@ruc.edu.cn; Y. Song and X. Zang, Kuaishou Technology Co., Ltd., No. 6 Shangdi West Road, Haidian District, Beijing,100085, China; emails: yangsong@kuaishou.com, zangxiaoxue@kuaishou.com.

1046-8188/2023/04-ART111 $15.00
https://doi.org/10.1145/3582425

1 INTRODUCTION

With overwhelming information from the Internet, recommender systems and search engines are two prevailing approaches for users to filter and retrieve the information they are interested in. In the past decades, recommendation and search services were traditionally deployed as two separate systems, serving different users with different information objectives. In recent years, there has been a tendency for content platforms to provide search and recommendation services simultaneously. Meanwhile, heterogeneous user behaviors from two services can be connected via the same sets of users and items. Early studies showed that jointly optimizing both recommendation and search models can lead to improvements in their respective performances [48, 49].

Traditional recommender systems assume that user historical behaviors contain user interests and leverage interacted item sequences to capture user preferences [35, 38, 56]. However, the user-item interaction data is affected by many factors, such as the exposure mechanism, sales promotion, and display position. Many existing models embed the rich information from the user, the item, and the context into real-value embedding vectors and make predictions using these vectors, e.g., using dot product between the user and item embeddings. From the perspective of causal analysis, the signals characterized by the embeddings can be decomposed into two parts: the causal association part, which describes why a user prefers an item under the context, and the non-causal association part, which reflects other factors influencing the user-item interaction. Thus the causal part reflects the true user preference, and the non-causal part reflects the statistical dependencies between the user-item pair and feedback. The striking differences between causal and non-causal associations lead to their distinctive roles in recommender systems. On the one hand, the causal association part predominantly contributes to the outcomes (e.g., clicks); on the other hand, the non-causal association part still influences the outcomes via the unobserved confounders. In order to attain optimal performance, an ideal recommender system should handle these two associations with different methods. Existing recommender systems, however, mostly ignored the differences between the two parts and used the embeddings as a whole.

Fortunately, the fact that search and recommendation services are deployed in the same content platform provides us with a chance to address this problem. The key difference between user search and recommendation activities is that users issue queries in search scenarios. A conspicuous feature of the search queries is that users actively issue their queries, which means that queries are composed of user intents or interests, which should not be affected by the confounders in the recommender, e.g., exposure mechanisms. Therefore, the user search queries can be employed to distinguish the above-mentioned two associations in the embeddings of the recommender.

In this article, we propose a model-agnostic framework IV4Rec+ that can effectively decompose the embedding vectors into two parts by jointly considering user behaviors in search and recommendation. Specifically, adopting the concepts in causal analysis, the search queries are employed as ***instrumental variables*** **(IVs)** to decompose original embedding vectors in recommendation, i.e., *treatments*, into two parts. Queries are embedded by pre-trained language models and then regressed on treatments. The fitted part is unrelated to the confounders because IVs (queries) are not affected by confounders. The residual part reflects the confounders because the residual is obtained by subtracting treatments from the fitted part. These two parts correspond to the causal association and non-causal association, respectively. Considering that the causal and non-causal parts both contribute to the final outcome, we reconstruct treatments using these two parts to mine their different roles in prediction by combining them with different weights learned by deep neural networks. Finally, the reconstructed treatment vectors are fed into the recommendation models for making the final prediction.

In causal inference, IVs methods have been widely used to learn the cause-effects between confounded treatment and outcome variables [12, 42]. After identifying IVs, which affect the outcome

only through treatment and not the confounders, the IV regression can estimate the causal relationship by first regressing treatment on the instrument, and then regressing the outcome on the treatment conditioned on the instrument. Treatment variables can be split into two parts: the fitted value in regression, which has a causal correlation, and the residual in regression, which has a non-causal correlation. In recent years, more and more platforms have deployed their search and recommendation together, serving the same set of users and items. Though issued in the search scenario, the queries reflect the user's true preference in both search and recommendation scenarios. Therefore, it is plausible to utilize user search activities as IVs.

Compared to existing IVs methods, the proposed IV4Rec+ has two striking differences. First, existing IVs methods focus on identifying cause-effects, while the goal of IV4Rec+ is to boost prediction performance. Therefore, IV4Rec+ applies a multi-task schema, including the recommendation prediction task and the causal learning task, to inject queries as IVs into recommendation models. Second, original IVs methods only use the fitted part of IVs regression to estimate the causal association between treatments and outcomes. To obtain better performance, IV4Rec+ leverages both the fitted and residual parts to make predictions.

Our main contributions are summarized as follows:

— We propose a model-agnostic causal learning framework, IV4Rec+, which leverages search queries to enhance the recommendation models. By considering search queries as IVs, we inject user search activities into the learning of the recommendation model in a causal learning manner. Moreover, an end-to-end multi-task learning schema is developed to learn the model parameters.
— We propose two variants of IV4Rec+, which use search queries to decompose different embeddings in recommendation. One decomposes item embeddings, and another decomposes both user and item embeddings.
— We conduct extensive experiments on two real-world industrial datasets and a public benchmark dataset. Experimental results validate that the proposed IV4Rec+ can consistently enhance different recommendation models. In addition, IV4Rec+ outperforms existing SOTA approaches which assist recommendation with search.

This article is an extended version of our previous work [27] published at the WWW 2022. We expand the IV4Rec framework proposed in the original article to the IV4Rec+, with major extensions: (1) We propose an end-to-end multi-task learning schema to train models with the recommendation task and causal learning task. The original IV4Rec solves the IV regression with analytical solutions. To obtain better performance, IV4Rec+ conducts the regression with deep neural networks and is optimized with gradient descent. (2) In IV4Rec+, we develop two strategies to incorporate search queries as IVs for users and items, called IV4Rec+(UI) and IV4Rec+(I). The two strategies make IV4Rec+ more flexible in real-world applications. (3) We conduct statistical analysis to verify the feasibility of using search queries as IVs for recommendation. Besides, we illustrate the causal graphs with details and discuss the feasibility of using search queries as IVs. (4) We conduct more experiments and analyses on a larger real-world industrial dataset and apply IV4Rec+ over an extra underlying model. All of the experiments demonstrate the effectiveness of IV4Rec+.

2 RELATED WORK

Generally, this article is highly related to three research lines: the joint modeling of search and recommendation, IVs regression, and causal learning for recommendation. We briefly review these research fields in this section.

2.1 Joint Modeling of Search and Recommendation

On par with search engines, recommender systems are widely deployed to help users filter the overwhelming information, alleviating the information overload issue. Traditionally, researchers and practitioners design search and recommendation services as two distinct systems. Search engines [8, 19, 21] and recommender systems [5, 10, 24, 43] have been developed under the design. Both services are ubiquitous in modern society. Early studies [11, 41] pointed out that search (information retrieval) and recommendation (information filtering) are the two sides of the same coin. These two services share the common goal: providing users with information objects to match their needs (which may or may not explicitly contain a query). The key difference between them is whether the users issue queries.

Unified recommendation and search model. Motivated by the similarities and connections between recommendation and search, unified models have been proposed. In e-commerce, an early work [32] designs a unified recommendation and search system to utilize complementary information from two tasks by integrating recommendation features and search features. A unified framework using deep convolution neural networks [26] was deployed at Flipkart, India's largest e-commerce company, to serve for visual search and recommendation. Zamani and Croft [48] assume that search and recommendation models could potentially benefit from each other. And a joint learning framework is proposed to train these two models simultaneously by optimizing a joint loss. Zamani and Croft [49] extend this work into a multi-task framework that learns a retrieval model from user-item interaction data and reconstructs item description texts that can be used for item retrieval. Yao et al. [45] design an approach called USER that mines user interests from the integrated user behavior sequences and accomplishes these two tasks in a unified way, alleviating the data sparsity problem and improving user satisfaction in both tasks.

One service facilitating the other. Since there exist heterogeneous user behaviors in two services, it is reasonable to incorporate behaviors in one service to improve the performance of the other. An early work [46] leverages search keywords of new users to address the cold-start problem in recommendation. Wu et al. [35] have proposed an approach to combining search history and browsing history logs to enhance the recommendation task. Wu et al. [40] propose a zero-shot heterogeneous transfer learning framework that transfers the learned knowledge from recommendation model to the search engine and addresses the cold-start problem in search. In this article, we also use the search data as external information to enhance the recommendation model.

2.2 Causal Learning for Recommendation

Recently, the causal inference [23] has been adapted to recommendation. Many researchers focus on causal embedding for recommendation [3, 14, 37, 55]. They are interested in finding the optimal treatment recommendation policy that maximizes the reward concerning the control recommendation policy for each user [3] or learning a fair or unbiased representation of items and users for recommendation [14, 37, 55]. Many recent works leverage the causal graph [23], an effective tool to depict cause effects and to identify the effect of bias in recommendation. A few studies [33, 51–53, 55] demonstrate the mechanism of biases, e.g., popularity bias and exposure bias, with the causal graph. Liu et al. [20] illustrate the generation process of biased and unbiased user-item interactions via two causal graphs and then mitigate the bias through an information bottleneck approach. Other researchers [34, 44] formulate the procedure of user behaviors in a causal inference framework and boost the model performance with data augmentation. Zhang et al. [50] propose a counterfactual method to learn accurate and robust user representation. Currently, some researchers [53–55] utilize user conformity and item popularity to facilitate recommendation models, which reveals that biases are not always harmful and can be beneficial with tailored usage. This article integrates the causal and non-causal relationships to improve the model performance.

2.3 IVs Regression

The IVs regression [4, 7, 12, 31] is a popular method in causal inference, which has been widely applied in statistics, econometrics, and epidemiology. This method allows for (unobserved) confounding bias and leverages external information as IVs to identify the causal relationship between treatment and outcome variables [23]. **Two-stage least squares** (**2SLS**) procedure [18] is a representative method for instrumental variable regression with linear models. Many researchers have explored this method in machine learning. Venkatraman et al. [31] adapt IVs regression to online learning which can update the estimator with streaming data instead of the whole dataset. McCulloch et al. [22] develop a flexible framework for IVs regression which uses Bayesian Additive Regression Trees in machine learning to serve for IVs model. IVs regression is also capable of assisting reinforcement learning. The offline policy evaluation in reinforcement learning can be improved with IVs to address the confounding bias in Q-function estimates [6]. With the development of deep learning methods, many recent works extend 2SLS with deep neural networks, which means the regression procedure can be non-linear and high-dimensional. Hartford et al. [12] propose a remarkable work to estimate cause-effect using IVs through deep learning techniques. Singh et al. [28] design a single-stage kernel-based IVs method that relaxes linear assumptions and offers a theoretical guarantee under mild assumptions. Xu et al. [42] provide an alternating training regime to combine 2SLS and deep learning methods and attain good end-to-end performance in high-dimensional image data and off-policy reinforcement learning tasks. Yuan et al. [47] utilize mutual information to automatically learn representations of IVs and confounder variables, which are used as inputs for 2SLS with neural network structure. In this article, we utilize IVs regression with deep neural networks.

3 PROBLEM FORMULATION

This section formalizes the problem of recommendation with search queries as IVs.

3.1 Preliminaries

3.1.1 Recommendation and Search in One Platform. In recent years, many content platforms have provided both search and recommendation services, which serve the same set of users with the same set of items. Thus user search and recommendation activities can be connected through a common set of users or items.

From the viewpoint of recommendation, when a user $u \in \mathcal{U}$ accesses the platform, the system provides a list of items $i \in \mathcal{I}$ with an existing recommendation model, where $\mathcal{U}$ and $\mathcal{I}$ denote the sets of all users and items respectively. Usually, a user u interacts with items in certain contexts denoted as $\mathbf{p}_u$, including the user profile, search history, or situational context, which can be collected by the platform and represented as real-valued vectors (embeddings) $\mathbf{p}_u \in \mathbb{R}^{d_c}$, where d_c is the dimension of embedding for context. Usually, each user u and each item i can also be represented as real-valued vectors (embeddings), denoted as $\mathbf{t}_u \in \mathbb{R}^{d_u}$ and $\mathbf{t}_i \in \mathbb{R}^{d_i}$, respectively, where d_i and d_u are the dimensions of the embeddings for users and items. The recommender system is usually trained with the collected historical user-system interactions $\mathcal{D}^{\text{rec}}$ where each tuple $(u, i, c) \in \mathcal{D}^{\text{rec}}$ means that the item i was shown to the user u and the interaction is $c \in \{0, 1\}$ where $c = 1$ means click and $c = 0$, otherwise.

From the viewpoint of search, when a user $u \in \mathcal{U}$ issues a query $q \in \mathcal{Q}$ where q is a text query and $\mathcal{Q}$ is the set of all queries, the system also provides a list of items $i \in \mathcal{I}$ with an existing search model. Similarly, each query can be represented as an embedding vector $\mathbf{t}_q \in \mathbb{R}^{d_q}$, where d_q is the embedding dimension. Since search and recommendation shared the same set of users $\mathcal{U}$ and items $\mathcal{I}$, the user u and item i in search are also represented as the same embeddings $\mathbf{t}_u$ and $\mathbf{t}_i$

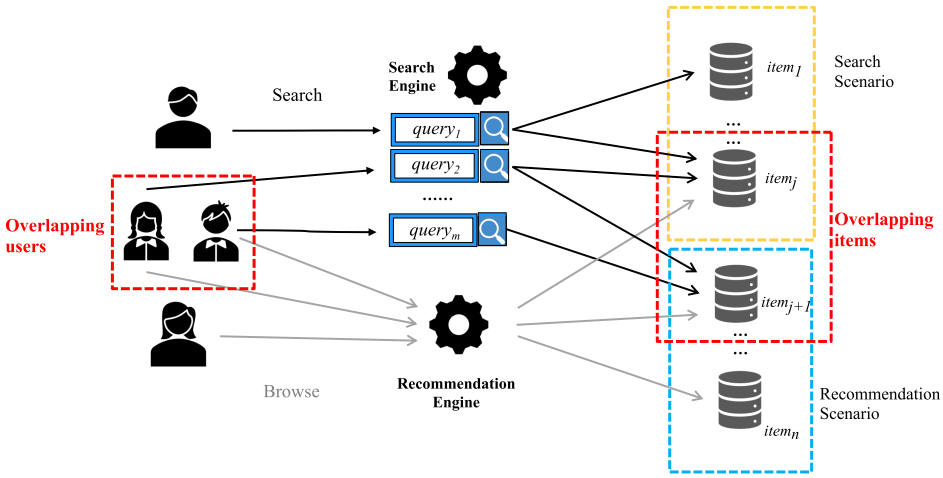


Fig. 1. Recommendation and search services in one platform. Search scenario: users issue queries and click returned items. Recommendation scenario: users browse returned items. There exist overlapping items and users in both services.

which are identical to those in recommendation. The historical user-system interactions in search can be denoted as $\mathcal{D}^{\text{src}}$ where each tuple $(u, q, i, c) \in \mathcal{D}^{\text{src}}$ indicates that a user u is shown with item i after issuing the query q, and the user activity is $c \in \{0, 1\}$.

Since the search and recommendation serve the same set of users with the same set of items, there inevitably exist overlaps between $\mathcal{D}^{\text{rec}}$ and $\mathcal{D}^{\text{src}}$. As shown in Figure 1, there exist overlapping target items in search and recommendation scenarios. That is, many items occur in both search and recommendation records. Besides, there also exist many overlapping users who have both search history and browsing history. These two phenomena provide us a chance to enhance recommendation using search data.

3.1.2 Method of IVs. In causal inference, IVs methods [1, 7] estimate the causal effect between a treatment variable X and an outcome variable Y in the presence of other variables (e.g., confounders) that simultaneously affect the treatment and outcome. Theoretically, a variable Z is a valid *instrumental variable* if the confounder (may be unobserved) unconfounds it and only affects the outcome Y via the treatment X. Typical IVs methods such as 2SLS [18] adopt a two-stage least square regression to identify causal effects of treatment X on outcome Y. First, regress the treatment on the instrument and obtain a reconstructed treatment; then, regress the outcome on the reconstructed treatment from the first stage. Intuitively, the reconstructed treatment is not affected by confounders since it is transformed from a confounder-free instrument. Then an unbiased estimate of cause-effect can be achieved from the coefficients of the second-stage regression.

3.2 Causal View of Recommendation

The goal of recommendation models is to capture the user interest and intention in the analysis of the user-item interactions so as to recommend satisfactory items. Existing recommender systems are usually trained on the user-system historical activities $\mathcal{D}^{\text{rec}}$, assuming that the click c in each of the training records $(u, i, c) \in \mathcal{D}^{\text{rec}}$ unbiasedly reflects the preference of u to i. From the perspective of causal inference, conventional recommendation models consider the user-item pair (u, i) as the treatment variable and the user feedback, i.e., the click c, as the outcome variable. Ideally, the cause-effect between them only consists of the preference of u to i. In the real

world, however, many confounders may affect the user clicks recorded in $\mathcal{D}^{\text{rec}}$, including the exposure mechanism, display position, sales promotion, and so on. Thus the associations between treatments and outcomes are a mixture of causal and non-causal relations.

Following the framework in [23], a causal graph for conventional recommender systems can be constructed in Figure 2(a), where the **treatment variable** (u, i) is denoted as $\mathcal{T}_{u,i}$ and represented by the corresponding embedding feature of user u and item i, the **outcome variable** c is denoted as $Y_{u,i}$, i.e., $Y_{u,i} = 1$ if a click event happens, $Y_{u,i} = 0$ otherwise. The **causal association** denotes the user u's true preference on item i leading to a user-item interaction. The **non-causal association** denotes the confounding factors, e.g., item display position, leading to a user-item interaction. The "non-causal" here means not related to the user preferences. Conventional recommender systems simply estimate the probability of potential user-item interactions by fitting models from treatments $\mathcal{T}_{u,i}$ to outcomes $Y_{u,i}$, resulting in learning mixed associations. Due to the presence of (unknown) confounders B, there exist two paths from treatment $\mathcal{T}_{u,i}$ to outcome $Y_{u,i}$, including a path of non-causal association that is facilitated by the confounder (the red arrow curve from $\mathcal{T}_{u,i}$ to $Y_{u,i}$), and a path of causal association that describes why an item is preferred by a user (the blue arrow line from $\mathcal{T}_{u,i}$ to $Y_{u,i}$). For instance, the reason why a user u clicked an item i can be that u was a little bit interested in i and i was displayed just right at the highest position in the list. The higher position makes (u, i) pair more likely to occur in $\mathcal{D}^{\text{rec}}$ because users need to scroll down to discover items at the lower position. Thus, position affects the occurrence of user-item pairs, i.e., the treatments. Meanwhile, a higher position leads to a higher probability of click [2], i.e., $c = 1$. Thus position also affects the outcome. Non-causal and causal associations reflect different relations between user-item pairs (i.e., treatments) and user feedback (i.e., outcomes).

It is difficult to identify the causal associations based on the biased observations $\mathcal{D}^{\text{rec}}$ given the unknown number of unknown confounders. Fortunately, the user search activities in $\mathcal{D}^{\text{src}}$ provide us a chance to decompose the treatment $\mathcal{T}_{u,i}$. As shown in Figure 2(b), we leverage the related queries as IVs, denoted as $\mathcal{Z}_{u,i}$, and regress $\mathcal{T}_{u,i}$ on $\mathcal{Z}_{u,i}$ to get $\widehat{\mathcal{T}}_{u,i}$ which does not depend on the confounders B. Thus the relation between $\widehat{\mathcal{T}}_{u,i}$ and $Y_{u,i}$ can be seen as a causal association. We also calculate the residuals $\widetilde{\mathcal{T}}_{u,i}$ of the regression. The relation between $\widetilde{\mathcal{T}}_{u,i}$ and $Y_{u,i}$ can be seen as a non-causal association. Treatments are reconstructed by combining the fitted vectors $\widehat{\mathcal{T}}_{u,i}$ and the residuals $\widetilde{\mathcal{T}}_{u,i}$. Therefore, user search activities are injected into recommendation under a causal learning framework, where both causal and non-causal associations are disentangled and leveraged for prediction in different manners.

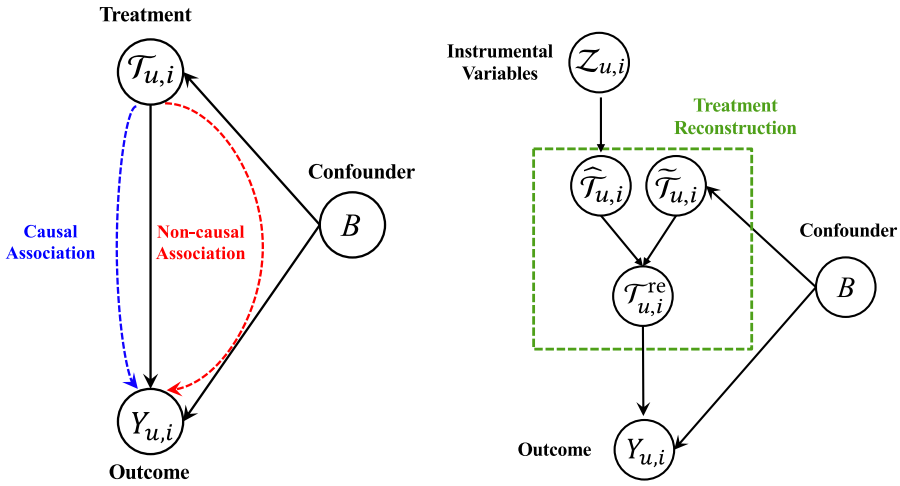
4 OUR APPROACH: IV4REC+

In this section, we first present the general IV4Rec+ framework, elaborating on how to enhance recommendation with search queries. We then demonstrate how to use search queries for items and users with details. We propose two strategies to leverage search queries as IVs to explore the potential of search activities for recommendation. Lastly, we present the model training procedure.

4.1 General Framework

To capture different impacts of causal and non-causal associations, we adopt user search queries as IVs to decompose original representations of the user-item pair (u, i), as shown in Figure 3.

Conventional recommender systems usually model users and items as latent vectors. The treatment variable $\mathcal{T}_{u,i}$ in recommender systems can be defined as a set of embeddings of the target item i and the user u. The recommendation models can make predictions based on these embeddings. Let f_{pred} denote the prediction module. The conventional recommender systems directly take $\mathcal{T}_{u,i}$ as input for f_{pred} to estimate clicks $Y_{u,i}$, neglecting the striking differences between causal and



(a) Conventional recommender systems mix the causal and non-causal associations between treatments and outcomes.

(b) IV4Rec+ disentangles the causal and non-causal parts explicitly with the help of search data and mines their different roles in prediction.

Fig. 2. The black parts of these figures are causal graphs for conventional recommender systems and our framework. The colored parts are our annotations. (a): conventional recommender systems. (b): recommender systems intervened by IVs. $\mathcal{T}_{u,i}$: embeddings of user and item. B: confounders (e.g., position bias). $Y_{u,i}$: user feedback. $\mathcal{Z}_{u,i}$: IVs (i.e., queries). $\widehat{\mathcal{T}}_{u,i}$: the fitted vectors. $\widetilde{\mathcal{T}}_{u,i}$: the residuals. The *causal association* denotes the user u's true preference on item i leading to a user-item interaction, i.e., cause-effects flowing from $\mathcal{T}_{u,i}$ to $Y_{u,i}$. The *non-causal association* denotes the confounding factors, e.g., item display position, which lead to a user-item interaction but are facilitated by confounders B and not user preferences. The *treatment reconstruction* means IV4Rec+ decomposes treatments into two parts and uses them to reconstruct treatments.

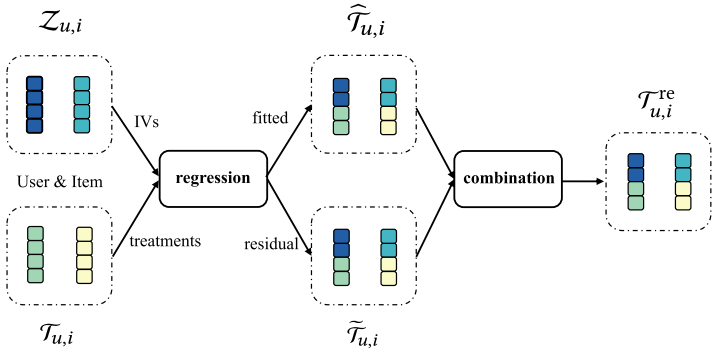


Fig. 3. General framework of IV4Rec+: the treatment reconstruction via using queries as IVs. Treatments denote embeddings of users and items in recommendation. By regressing treatments on IVs, we can get the fitted part and the residual part representing the causal and non-causal associations, respectively. Then we combine them with different weights to reconstruct treatments.

non-causal associations, as shown in Figure 2(a). To address this problem, the proposed framework IV4Rec+ first retrieves search queries corresponding to each user and item to serve as IVs, denoted as $\mathcal{Z}_{u,i}$. According to the attributes of IVs (i.e., IVs are unconfounded by the confounder and only affect the outcome via treatment), we regress user and item embeddings $\mathcal{T}_{u,i}$ on query

embeddings $\mathcal{Z}_{u,i}$ to achieve the *fitted part* $\widehat{\mathcal{T}}_{u,i}$, which reflects the causal association in the recommender system and doesn't depend on the confounders. After getting the fitted vectors in $\widehat{\mathcal{T}}_{u,i}$, we also get the *residual part* of the regression $\widetilde{\mathcal{T}}_{u,i}$, which contains the non-causal association in the recommender system and reflects the impacts of confounders. To better use both associations, $\widetilde{\mathcal{T}}_{u,i}$ and $\widehat{\mathcal{T}}_{u,i}$ are assigned different weights to control their impacts on the final prediction. Then the reconstructed treatment $\mathcal{T}^{\text{re}}_{u,i}$ is achieved by combining $\widetilde{\mathcal{T}}_{u,i}$ and $\widehat{\mathcal{T}}_{u,i}$. Finally, the click is predicted based on reconstructed user and item representations. The whole procedure follows the causal graph in Figure 2(b).

By considering there exist both overlapping items and users in search and recommendation, we propose two strategies to utilize search queries as IVs. We denote the representations of (u, i) in existing sequential recommendation models as $\mathbf{t}_u$ and $\mathbf{t}_i$, where $\mathbf{t}_u$ and $\mathbf{t}_i$ are embedding vectors for user u and item i, respectively. Intuitively, we can directly take queries issued by the user u to serve as his IVs and use queries clicked the item i to serve as its IVs. Besides, the user embedding $\mathbf{t}_u$ is often calculated by aggregating the user's historically interacted items. Thus, it is also plausible to collect queries clicked user's interacted items to serve as the user's IVs.

4.2 IV4Rec+(UI)

We first present IV4Rec+(UI), which takes the queries issued by the user u to serve as u's IVs, and uses the queries clicked the item i to serve as i's IVs.

4.2.1 Construction of Treatments and IVs. The treatment variable $\mathcal{T}_{u,i}$ in recommender systems represents the user u and item i, which can be expressed as a set of two vectors:

$$\mathcal{T}_{u,i} = \{\mathbf{t}_u, \mathbf{t}_i\}, \tag{1}$$

where the item embedding $\mathbf{t}_i$ is usually obtained by some representation learning methods that project features (e.g., content) into a dense vector. For sequential recommendation, the user embedding $\mathbf{t}_u$ is usually learned by aggregating the user historically interacted items [16], as well as other information such as user profile [56] and search history [35]. The treatment construction of IV4Rec+(UI) differs from IV4Rec of the conference article in terms of user embedding $\mathbf{t}_u$, where IV4Rec+(UI) considers the aggregated vectors of user histories as treatments.

The IVs for treatment $\mathcal{T}_{u,i}$, denoting $\mathcal{Z}_{u,i}$, are defined as a set of two vectors:

$$\mathcal{Z}_{u,i} = \{\mathbf{z}_u, \mathbf{z}_i\}, \tag{2}$$

where $\mathbf{z}_i$ and $\mathbf{z}_u$ are the IVs for $\mathbf{t}_i$ and $\mathbf{t}_u$, respectively. They are constructed as follows.

As for IV $\mathbf{z}_i$ for the item i, we first recall the corresponding query q, which clicked item i in $\mathcal{D}^{\text{src}}$, where $q : (u', q, i, c = 1) \in \mathcal{D}^{\text{src}}$ is the recalled query. And then, query embedding $\mathbf{z}_i$ can be generated by applying pre-trained language models (e.g., BERT [9]) to the most recently issued query q.

As for IV $\mathbf{z}_u$ for the user u, similar to the item i, we aggregate the search history into $\mathbf{z}_u$ to serve as IVs for user recommendation history embedding $\mathbf{t}_u$. Denoting Q_u as the set of queries in the user u's search history, i.e., $Q_u = \{q : (u, q, i', c') \in \mathcal{D}^{\text{src}}\}$. These queries are also embedded by pre-trained language models. For $q_j \in Q_u$, its embedding is denoted as $\mathbf{z}_j$. Then $\mathbf{z}_u$ can be calculated by first feeding these query embeddings in a multi-head self-attention network for learning their contextual representations and then combining these interacted query representations with an additive attention network. The procedure is denoted as f_{search}. Specifically, the IV $\mathbf{z}_u$ for the user u is defined as

$$\mathbf{z}_u = f_{\text{search}}(Q_u) = \sum_{k=1}^{|Q_u|} \beta_k \mathbf{e}_k, \tag{3}$$

where $\mathbf{e}_k$ is the representation of the kth query in Q_u, which is a concatenation of the representation from a multi-head self-attention network with H heads:

$$\mathbf{e}_k = \mathbf{W}^O \text{Concat}(\mathbf{e}_{k,1}, \mathbf{e}_{k,2}, \ldots, \mathbf{e}_{k,H}),$$

where $\mathbf{W}^O \in \mathbb{R}^{d_q \times H d_k}$ is the projection matrix, and $\mathbf{e}_{k,h}$ is the representation of the kth query learned by hth attention head:

$$\mathbf{e}_{k,h} = \mathbf{W}_h^V \left(\sum_{j=1}^{|Q_u|} \alpha_{k,j}^h \mathbf{z}_j \right),$$

where $\alpha_{k,j}^h = \frac{\exp((\mathbf{W}_h^Q \mathbf{z}_k)^\intercal \mathbf{W}_h^K \mathbf{z}_j)}{\sum_{m=1}^{M} \exp((\mathbf{W}_h^Q \mathbf{z}_k)^\intercal \mathbf{W}_h^K \mathbf{z}_m)}$, $\mathbf{W}_h^Q$, $\mathbf{W}_h^K$, and $\mathbf{W}_h^V \in \mathbb{R}^{d_k \times d_q}$ are the projection parameters in the hth self-attention head, $\alpha_{k,j}^h$ is the weight indicating importance of the interaction between kth and jth queries, and $\mathbf{z}_j$ is the query embedding of q_j and $q_j \in Q_u$.

As for the attention weights β in Equation (3), the β_k for the kth query is computed as

$$\beta_k = \frac{\exp(\hat{\beta}_k)}{\sum_{j=1}^{|Q_u|} \exp(\hat{\beta}_j)},$$

where $\hat{\beta}_k = \mathbf{w}^\intercal \tanh(\mathbf{V}_1 \mathbf{q}_k + \mathbf{V}_2 \mathbf{e}_k + \mathbf{b})$, $\mathbf{V}_1 \in \mathbb{R}^{d_k \times d_{q'}}$, $\mathbf{V}_2 \in \mathbb{R}^{d_k \times d_q}$, and $\mathbf{w}, \mathbf{b} \in \mathbb{R}^{d_k}$ are parameters in the attention module. The vector $\mathbf{q}_k \in \mathbb{R}^{d_{q'}}$ represents the query "Q" in the attention mechanism. For example, $\mathbf{q}_k$ can be set as $\mathbf{e}_k$. In order to learn qualified IVs for user embedding and apply our framework flexibly, we customize $\mathbf{q}_k$ in several ways to adjust the aggregation of search history to different user modeling modules, as will be discussed in Section 4.2.3. For IVs construction, IV4Rec+(UI) differs from IV4Rec in terms of considering the aggregated vectors of search histories as IVs.

4.2.2 Treatment Reconstruction. Based on the original treatment $\mathcal{T}_{u,i}$ and IVs $\mathcal{Z}_{u,i}$, we show that a new treatment $\mathcal{T}_{u,i}^{\text{re}}$ can be created by first regressing $\mathcal{T}_{u,i}$ on $\mathcal{Z}_{u,i}$ and then combining the fitted vectors and the residuals. Through IVs regression, we can decompose the original treatments $\mathcal{T}_{u,i}$ into the *fitted part* $\widehat{\mathcal{T}}_{u,i}$ and the *residual part* $\widetilde{\mathcal{T}}_{u,i}$, reflecting the user preference and other confounding factors, respectively. Considering that both the fitted and residual parts contribute to the outcome prediction, we utilize both parts by combining them with different weights to tailor their impacts.

Treatment decomposition. The goal of treatment decomposition is to disentangle the causal and non-causal associations between treatments and outcomes. The IVs method is leveraged to isolate the causal association flowing from the treatments to outputs.

In IV4Rec+(UI), IVs regression is conducted for users and items separately. Specifically, we estimate the causal part by projecting the treatments onto the IVs:

$$\widehat{\mathcal{T}}_{u,i} = \left\{ \hat{\mathbf{t}}_u = f_{\text{proj}}^u(\mathbf{z}_u), \hat{\mathbf{t}}_i = f_{\text{proj}}^i(\mathbf{z}_i) \right\}, \tag{4}$$

where $\hat{\mathbf{t}}_u \in \mathbb{R}^{d_u}$ and $\hat{\mathbf{t}}_i \in \mathbb{R}^{d_i}$ denote the fitted vectors. f_{proj}^u and f_{proj}^i are the treatment regression networks, both implemented as two **Multi-layer Perceptrons** (**MLPs**) to map IVs to treatments. The IVs methods usually apply a mean square error loss function to supervise the regression from the instrument to the treatment. We also introduce this loss function and leave the details in Section 4.4.

After getting the fitted vectors in $\widehat{\mathcal{T}}_{u,i}$, it is easy to get the *residual parts* of the regression $\widetilde{\mathcal{T}}_{u,i}$:

$$\widetilde{\mathcal{T}}_{u,i} = \{ \tilde{\mathbf{t}}_u = \mathbf{t}_u - \hat{\mathbf{t}}_u, \tilde{\mathbf{t}}_i = \mathbf{t}_i - \hat{\mathbf{t}}_i \}, \tag{5}$$

which contains the non-causal association in recommendation.

Treatment combination. To better predict the outcomes, we leverage both the fitted vectors $\widehat{\mathcal{T}}_{u,i}$ and the residuals $\widetilde{\mathcal{T}}_{u,i}$ to capture their associations with outcomes. Due to their different roles in user-item interaction, we combine them with adaptive weights, which determine the importance of each part to aggregate $\widehat{\mathcal{T}}_{u,i}$ and $\widetilde{\mathcal{T}}_{u,i}$.

The fitted vectors $\widehat{\mathcal{T}}_{u,i}$ and the residuals $\widetilde{\mathcal{T}}_{u,i}$ can be combined, achieving a reconstructed treatment $\mathcal{T}^{\text{re}}_{u,i}$:

$$\mathcal{T}^{\text{re}}_{u,i} = \left\{ \mathbf{t}^{\text{re}}_u = \hat{\mathbf{t}}_u + \alpha_u \tilde{\mathbf{t}}_u, \mathbf{t}^{\text{re}}_i = \hat{\mathbf{t}}_i + \alpha_i \tilde{\mathbf{t}}_i \right\}, \tag{6}$$

where $\alpha_u, \alpha_i \in [0, 1]$ are weights for the residuals of user u and item i. These two weights are, respectively, estimated as

$$\alpha_u = f^u_{\text{weight}}(\text{Concat}(\hat{\mathbf{t}}_u, \mathbf{t}_u)), \quad \alpha_i = f^i_{\text{weight}}(\text{Concat}(\hat{\mathbf{t}}_i, \mathbf{t}_i)),$$

where f^u_{weight} and f^i_{weight} denote the two estimators for users and items, respectively. They are also implemented as two MLPs, with sigmoid functions applying to the output layer. The inputs of the MLPs are concatenations of the treatments and the fitted values corresponding to (u, i). In this way, α_u and α_i are estimated as weights to scale down the impacts of the residuals based on the interactions between the fitted vectors and the treatments.

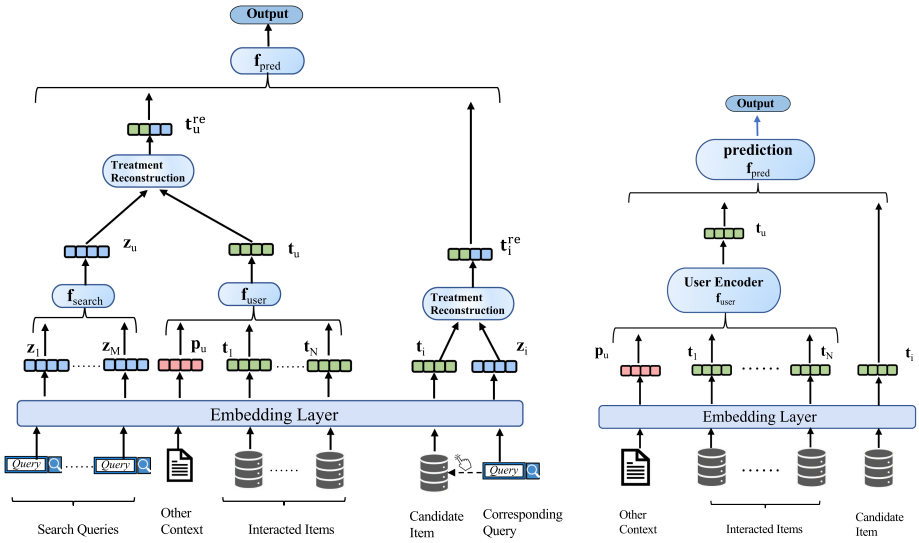
Please note that in traditional causal inference, researchers focus on identifying the cause-effects from observed data. Therefore, the residual part in IVs regression is often discarded to remove the non-causal association. This aligns well with their goal of just estimating the causal association, as shown in Figure 2(a). In the recommendation scenario of this article; however, our goal is to make accurate predictions rather than just identifying cause-effects. Since the residual part can still contribute to the user preference prediction via the red curve in Figure 2(a), the fitted part and the residual part are complementary in the prediction task. Recent studies [53–55] also argued that non-causal associations could enhance prediction accuracy. For instance, considering a specific confounder (the popularity bias), user conformity, and item popularity can help recommendation models make reasonable and accurate predictions. This motivates us that the residuals can be leveraged to improve the recommendation performance.

Also, note that α_i and α_u scale down the impacts of the residuals. That's because the causal association plays a predominant role in the click event, and the non-causal association contains trivial signals along with significant signals. Also, the confounding factors between each (u, i) pair and feedback c are multifarious. Thus, we use deep neural networks to learn adaptive weights to reconstruct different treatment variables.

4.2.3 Model-agnostic Application. Many recommender systems [35, 38, 56] share a similar structure, as shown in Figure 4(b). We call them the underlying models. An underlying model represents items as embedding vectors, utilizes user historical behaviors to learn user representations, and predicts click probability of (u, i) based on their learned representations. Our proposed IV4Rec+ is a model-agnostic framework that can be implemented over existing recommender systems that follow this structure.

As shown in Figure 4(a), the proposed IV4Rec+(UI) can be applied to the underlying models by adding additional modules without changing the original models. Specifically, after the embedding layer, we get vectors of items and queries for the (u, i) pair. As for the candidate item i, its treatment and IV are $\mathbf{t}_i$ and $\mathbf{z}_i$. As for the user u, we need further express her/his treatment $\mathbf{t}_u$ and IV $\mathbf{z}_u$ through historical items and queries. Let $\mathcal{I}_u$ denote the set of items that interacted with the user u in $\mathcal{D}^{\text{rec}}$:

$$\mathcal{I}_u = \{ i' : \exists (u, i', c = 1) \in \mathcal{D}^{\text{rec}} \}.$$



(a) IV4Rec+(UI). (b) Underlying models.

Fig. 4. (a): The application of IV4Rec+(UI) over underlying models. (b): The structure of underlying models. Treatment Reconstruction: this module is illustrated in Figure 3.

The underlying models often capture the users' interests from the interaction sequences. The user embedding in recommendation scenario is achieved through the user behavior module f_{user}:

$$\mathbf{t}_u = f_{\text{user}}(\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{|\mathcal{I}_u|}, \mathbf{p}_u),$$

where $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{|\mathcal{I}_u|}$ are item vectors in the set of interacted items $\mathcal{I}_u$, $\mathbf{p}_u$ is the representation of other context for user u and f_{user} is usually implemented as attention mechanisms [35, 56] or other complex structures, e.g., the graph neural network and attention network [38].

As shown in Equation (3), IV4Rec+(UI) calculates the user embedding in search scenario $\mathbf{z}_u$ through f_{search}. We leverage the multi-head self-attention mechanism to enhance query representations and learn informative user embedding by selecting important queries with the help of additive attention. Please note that the "Q" in additive attention is customized in several ways to adapt to different underlying models. The aim of f_{search} is to capture user interest from query logs which is compatible with the user embedding learned by f_{user} so as to construct qualified IVs. Due to the fact that underlying models use various approaches to conducting the aggregation of interaction sequences, f_{search} is adapted to use different vectors as "Q" in additive attention according to different underlying models. Specifically, the model DIN [56] uses the candidate item to mine the user interests from historical behaviors, the model SRGNN [38] uses the last-clicked item to control the weight of each item in the historical sequence for aggregation, and the model NRHUB [35] uses the historical items themselves to learn the importance of each item. Our framework IV4Rec+ follows them accordingly to aggregate the search query sequence, that is, when IV4Rec+ is applied over DIN, SRGNN, or NRHUB, "Q" of additive attention in f_{search} is set as the candidate item, the last-clicked item or the query sequence respectively.

After that, we can reconstruct the treatments $\mathcal{T}_{u,i}$ as shown in Section 4.2.2 and the reconstructed treatments $\mathcal{T}_{u,i}^{\text{re}}$ are used to make predictions:

$$\hat{y}_{u,i} = f_{\text{pred}}\left(t_u^{\text{re}}, t_i^{\text{re}}\right), \tag{7}$$

where f_{pred} is a prediction module, usually implemented as an MLP [56] or dot product [35, 39].

4.3 IV4Rec+(I)

In this section, we present IV4Rec+(I) which only utilizes queries corresponding to items as IVs. IV4Rec+(I) follows the architecture of IV4Rec+(UI) with major differences in using queries as IVs for users.

4.3.1 Construction of Treatments and IVs. The intuitive idea is that the user embedding $\mathbf{t}_u$ is achieved by aggregating embeddings of the user's historically interacted items. Therefore, reconstructing embeddings of all interacted items leads to reconstructed user embedding.

The treatment variable $\mathcal{T}_{u,i}$ in recommender systems can be defined as a set of embeddings, including the embedding of the target item i and the embeddings of the items interacted with u:

$$\mathcal{T}_{u,i} = \{\mathbf{t}_j : j \in \mathcal{I}_u \cup \{i\}\}, \tag{8}$$

where $\mathbf{t}_j \in \mathbb{R}^{d_i}$ is the representation of item j and $\mathcal{I}_u$ is the set of u's interacted items.

The corresponding IVs $\mathcal{Z}_{u,i}$ of treatment $\mathcal{T}_{u,i}$ are defined as a set of embeddings of search queries:

$$\mathcal{Z}_{u,i} = \{\mathbf{z}_j : j \in \mathcal{I}_u \cup \{i\}\}, \tag{9}$$

where each embedding $\mathbf{z}_j$ is the vector of a search query related to the item j. Specifically, $\mathbf{z}_j$ can be constructed as follows. We retrieve a query q clicked item j from $\mathcal{D}^{\text{src}}$[1]:

$$q : (u', q, j, c = 1) \in \mathcal{D}^{\text{src}},$$

Then we embed query q as vector $\mathbf{z}_j$ using pre-trained language models. Hence, the $\mathcal{Z}_{u,i}$ is composed of query vectors highly relevant to items in $\mathcal{T}_{u,i}$. The difference of IVs construction between IV4Rec+(I) and IV4Rec lies in the number of queries per item. This article only recalls one related query for each item.

4.3.2 Treatment Reconstruction.

Treatment decomposition. In this strategy, we regress each item on its corresponding query:

$$\widehat{\mathcal{T}}_{u,i} = \left\{\hat{\mathbf{t}}_j = f_{\text{proj}}(\mathbf{z}_j) : j \in \mathcal{I}_u \cup \{i\}\right\}, \tag{10}$$

where f_{proj} is the treatment regression network and implemented as an MLP similar to Equation (4). We also minimize the mean square error loss to supervise the learning of IVs regression $f_{\text{proj}}(\mathbf{z}_j)$ and we detail the loss in Section 4.4.

Similar to Equation (5), we can calculate the *residual part* by

$$\widetilde{\mathcal{T}}_{u,i} = \{\tilde{\mathbf{t}}_j = \mathbf{t}_j - \hat{\mathbf{t}}_j : j \in \mathcal{I}_u \cup \{i\}\}, \tag{11}$$

The intuition is that the original treatments are projected onto the subspace spanned by the columns of the IVs. In the light of attributes of IVs (i.e., IVs are unconfounded by the confounder and only affect the outcome Y via treatment X), the *fitted part* which doesn't depend on the confounders reflects the causal association, and the *residual part* which depends on the confounders contains the non-causal association.

[1]To ensure each item has a corresponding query, we recall the most relevant query q for item j if there is no query clicked item j in $\mathcal{D}^{\text{src}}$. More details can be found in the experiment.

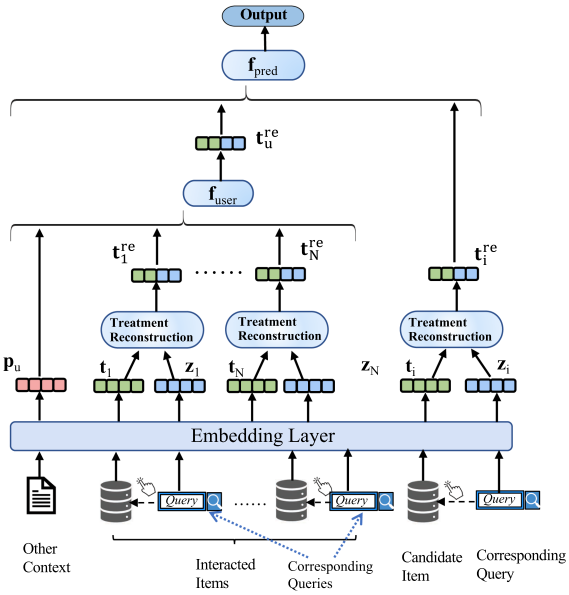


Fig. 5. The application of IV4Rec+(I) over underlying models. Treatment Reconstruction: this module is illustrated in Figure 3.

Treatment combination. Similar to Equation (6), we combine the fitted vectors $\widehat{\mathcal{T}}_{u,i}$ and the residuals $\widetilde{\mathcal{T}}_{u,i}$ for each item in $\mathcal{T}_{u,i}$:

$$\mathcal{T}_{u,i}^{\text{re}} = \left\{ \mathbf{t}_j^{\text{re}} = \hat{\mathbf{t}}_j + \alpha_j \tilde{\mathbf{t}}_j : j \in \mathcal{I}_u \cup \{i\} \right\}, \tag{12}$$

where $\hat{\mathbf{t}}_j \in \widehat{\mathcal{T}}_{u,i}$ and $\tilde{\mathbf{t}}_j \in \widetilde{\mathcal{T}}_{u,i}$ are vectors in these two sets, both correspond to the same item j, and the $\alpha_j \in [0, 1]$ is estimated by an MLP:

$$\alpha_j = f_{\text{weight}}(\text{Concat}(\hat{\mathbf{t}}_j, \mathbf{t}_j)),$$

where f_{weight} is an estimator implemented as an MLP and treats items in $\mathcal{I}_u$ and the item i identically.

4.3.3 Model-agnostic Application. In IV4Rec+(I), the user embedding is reconstructed by reconstructing all historically interacted items. Thus the IVs and treatments of the (u, i) pair are embedding vectors of queries and items, which can be obtained by the embedding layer. Then only the reconstruction module is injected after the embedding layer in underlying models to reconstruct all the item embeddings, illustrated in Figure 5.

Formally, representations of (u, i) in existing recommender systems are denoted as $\mathbf{t}_u$ and $\mathbf{t}_i$, where $\mathbf{t}_u$ can be calculated by f_{user}. After reconstructing items in treatments, we can obtain the reconstructed embedding for the candidate item and the user's interacted items. Based on these embeddings, we can get the reconstructed user embedding t_u^{re}:

$$\mathbf{t}_u^{\text{re}} = f_{\text{user}}\left(\mathbf{t}_1^{\text{re}}, \mathbf{t}_2^{\text{re}}, \ldots, \mathbf{t}_{|\mathcal{I}_u|}^{\text{re}}, \mathbf{p}_u\right), \tag{13}$$

where $\mathbf{t}_1^{\text{re}}, \mathbf{t}_2^{\text{re}}, \ldots, \mathbf{t}_{|\mathcal{I}_u|}^{\text{re}}$ are reconstructed item vectors in the set of interacted items $\mathcal{I}_u$, $\mathbf{p}_u$ is the representation of other context for user u and f_{user} is the original user modeling module in underlying models. Finally, the interaction probability is predicted based on learned user and item

representations:

$$\hat{y}_{u,i} = f_{\text{pred}}\left(t_u^{\text{re}}, t_i^{\text{re}}\right), \tag{14}$$

where f_{pred} is the prediction module.

4.4 Model Training

For IV4Rec+(UI), parameters in the proposed IV4Rec+ include parameters in f_{search}, f_{proj}^u, f_{proj}^i, f_{weight}^u, f_{weight}^i, and the parameters from the underlying recommendation model. For IV4Rec+(I), parameters in the proposed IV4Rec+ include parameters in f_{proj}, f_{weight}, and the parameters from the underlying recommendation model. All the trainable parameters are denoted as $\boldsymbol{\Theta}$ and trained based on $\mathcal{D}^{\text{rec}}$. The training procedure can be seen as a multi-task learning schema that applies additional supervision of IVs regression over the search representation and projection modules. Formally, the overall training loss is

$$\mathcal{L} = \mathcal{L}_O + \mathcal{L}_{\text{IV}} + \lambda \left\|\boldsymbol{\Theta}\right\|^2, \tag{15}$$

where $\left\|\boldsymbol{\Theta}\right\|^2$ is the regularization for avoiding over-fitting and λ is its coefficient. $\mathcal{L}_O$ and $\mathcal{L}_{\text{IV}}$ are the losses w.r.t. preference prediction task and causal learning (IV regression) task, respectively.

As for $\mathcal{L}_O$, the widely used binary cross entropy loss is adopted:

$$\mathcal{L}_O = -\frac{1}{|\mathcal{D}^{\text{rec}}|} \sum_{(u,i,c)\in\mathcal{D}^{\text{rec}}} c \cdot \log \hat{y}_{u,i} + (1-c) \cdot \log(1-\hat{y}_{u,i}), \tag{16}$$

where $\hat{y}_{u,i}$ is the predicted score for (u,i). This loss $\mathcal{L}_O$ is used to optimize all the trainable parameters to recover the historical interactions.

As for $\mathcal{L}_{\text{IV}}$, Sections 4.3.2 and 4.2.2 mentioned that we also apply additional supervision over $\widehat{\mathcal{T}}_{u,i}$ to ensure the fitted and the residual part represent the causal and non-causal parts, respectively. That is, we minimize the mean square error loss between the estimated $\widehat{\mathcal{T}}_{u,i}$ and the treatment $\mathcal{T}_{u,i}$ to guide the learning of IVs regression:

$$\mathcal{L}_{\text{IV}} = \begin{cases} \dfrac{1}{|\mathcal{D}^{\text{rec}}|} \displaystyle\sum_{(u,i,c)\in\mathcal{D}^{\text{rec}}} \gamma_1 \left\|\hat{\mathbf{t}}_u - \mathbf{t}_u\right\|^2 + \gamma_2 \left\|\hat{\mathbf{t}}_i - \mathbf{t}_i\right\|^2 & \text{IV4Rec+(UI)}, \\ \dfrac{1}{|\mathcal{D}^{\text{rec}}|} \displaystyle\sum_{(u,i,c)\in\mathcal{D}^{\text{rec}}} \sum_{j\in\mathcal{I}_u\cup\{i\}} \gamma_0 ||\hat{\mathbf{t}}_j - \mathbf{t}_j||^2 & \text{IV4Rec+(I)}, \end{cases} \tag{17}$$

where $\mathbf{t}_u, \mathbf{t}_i \in \mathcal{T}_{u,i}$, and $\hat{\mathbf{t}}_u, \hat{\mathbf{t}}_i \in \widehat{\mathcal{T}}_{u,i}$ are vectors in these two sets, and γ_1, γ_2 are hyper-parameters to balance the objectives for IV4Rec+(UI). The γ_0 is a hyper-parameter for IV4Rec+(I). They can be seen as the tradeoff parameters between the causal learning task and the prediction task.

Note that the idea of IVs regression methods is to regress treatments on IVs. The $\mathcal{T}_{u,i}$ is the "target variable" in this loss. Thus, the $\mathcal{T}_{u,i}$ is fixed during the IVs regression, which means we stop the gradients from the loss $\mathcal{L}_{\text{IV}}$ to the modules that generate treatments in underlying models. In our implementation, for IV4Rec+(UI), $\mathcal{L}_{\text{IV}}$ is used for updating $\hat{\mathbf{t}}_u, \hat{\mathbf{t}}_i \in \widehat{\mathcal{T}}_{u,i}$ according to the IVs regression methods. Therefore, it is minimized to optimize parameters in the search representation and treatment regression networks, i.e., parameters in f_{search}, f_{proj}^u, f_{proj}^i. For IV4Rec+(I), $\mathcal{L}_{\text{IV}}$ is minimized to optimize parameters in the treatment regression network, i.e., parameters in f_{proj}.

Compared to the original IV4Rec, the multi-task training schema can better balance the relationship between causal learning and recommendation tasks. Original IV4Rec solves the IV regression without considering the final recommendation task. Since our goal is to improve the recommendation performance, it is beneficial to learn the parameters of treatment regression networks under the supervision of the recommendation task. Moreover, with the introduction of deep neural networks, we cannot use analytical solutions in the original version to learn the parameters of neural networks. Thus it is necessary and effective to design a multi-task training schema.

We analyze the model size of IV4Rec+ to demonstrate the newly introduced model complexity. For IV4Rec+(I), the additional complexity is mainly caused by the treatment reconstruction in Section 4.3.2, where two new modules, i.e., two MLPs, including the f_{proj} and the f_{weight}, are introduced which contain about 100 k parameters. IV4Rec+(I) uses these modules to conduct $|\mathcal{I}_u|+1$ times treatment reconstruction, as stated in Equations (10)–(13). The additional complexity is determined by the computation of the MLPs and vector addition, which is much less than that of mechanisms used in sequential recommendation models, e.g., attention mechanisms. For IV4Rec+(UI), the additional complexity is caused by the IVs construction in Section 4.2.1 and treatment reconstruction in Section 4.2.2. IVs construction consists of a multi-head self-attention network and an additive attention network containing 500 k parameters. Treatment reconstruction consists of two MLPs, the f_{proj} and the f_{weight}, containing about 100 k parameters. We can observe that IVs construction has a comparable complexity to the backbone models because they use the similar attention mechanisms, while treatment reconstruction yields much less complexities of calculating the MLPs and vector addition for the users and the items. In addition, IV4Rec+(UI) and IV4Rec+(I) both introduce queries as IVs, which need to allocate more space to store the embedding layer of queries with size $d_q \cdot |\mathcal{Q}|$.

5 DISCUSSION

This section discusses the feasibility of using search queries as IVs and its difference from traditional IVs methods.

5.1 Feasibility of Using Search Queries as IVs

According to the theory of IVs estimations, the IVs have to satisfy two assumptions: exogeneity and relevance.

Service of recommendation. Also, users issue the queries actively which guarantees that queries only contain user intentions and interests. Therefore, the context of these search queries cannot be influenced by the ranking positions/exposure of the items in recommendation. That is, using search queries as IVs meets the assumption of exogeneity.

As for relevance, it means that the IVs (search queries) are the causes of the treatment, but do not directly affect the outcomes in recommender systems, i.e., the user click behavior. The search and recommendation share a common goal: providing users with items to satisfy their information needs. In search, the user information needs are explicitly summarized as queries. In recommendation, the information needs are implicitly summarized with the representations of users. When the search engines and recommender systems are deployed in one app and serve the same group of users with the same set of items, a large extent of search queries reflect part of the user information needs in recommendation. The phenomenon indicates that search queries can be seen as a cause of the treatment in recommendation. Considering that the IVs (search queries associated with an item and search queries issued by some users) are specific requests made by users in search, it is obvious that they do not directly affect the outcomes in recommendation. Furthermore, we provide statistical analysis to verify that search queries meet the relevance assumption in Section 6.4.1.

With the above analysis, we conclude that the embeddings of the search queries well satisfy the assumptions of exogeneity and relevance.

5.2 Difference with Traditional IVs Methods

In the field of causal inference, traditional IVs methods are powerful frameworks widely applied in econometrics, statistics, and epidemiology. They aim at identifying the cause-effect between treatments and outcomes, especially in the presence of unobserved confounders. Though inspired by the IVs methods, the proposed IV4Rec+ in this article is devoted to utilizing the causal

relationships between recommendation and search tasks to facilitate the recommendation model performance. Thus, the goal of IV4Rec+ is not the same as that of the traditional IVs methods. To achieve the goal, IV4Rec+ has made the following fundamental modifications for adapting the traditional method of IVs to recommendation:

(1) **Reconstruction of treatment with both causal and non-causal parts**: In traditional IVs methods, the residual of the IVs regression is discarded. In our approach, however, the residual is used as the embedding representation of the indirect non-causal association part. This is because our goal is not just to learn the causal associations. In the light of that the non-causal and causal associations both contribute to the final user feedback from different paths, they are complementary in the prediction task. Our experimental results also verify that finding a suitable reconstruction of these two parts from the original treatment is more helpful in enhancing the recommendation accuracy. On the other hand, removing the non-causal part leads to the reconstructed treatment $\mathcal{T}_{u,i}^{\text{re}}$ containing only $\widehat{\mathcal{T}}_{u,i}$ learned from search data, missing the signals from recommendation. That is, removing the non-causal part prevents the treatment module in recommendation from being updated with gradient descent. Thus, the whole model is estimating interactions according to search activities. It is no surprise that the recommendation performance will decrease drastically. The experimental results reported in Section 6.4.4 also verify the analysis.

(2) **IVs regression**: The classical IV regression usually consists of a two-stage procedure with two linear regressions. Inspired by recent studies [12, 28, 42, 47] which extend the two-stage linear least square with deep neural networks, we also apply the MLP to conduct the IVs regression. To achieve a favorable end-to-end training procedure, we develop a multi-task learning schema and introduce hyper-parameters to balance the causal learning task and prediction task. Due to the introduction of the neural networks and the multi-task learning schema to IV regression, we cannot theoretically prove the unbiasedness of the proposed frameworks. Traditional IV methods enjoy desirable theoretical properties, e.g., unbiasedness, since these methods simplify cause-effects by modeling all relationships as linear functions. Though several recent studies [12, 28, 42, 47] prove the unbiasedness of the IVs regression under specific deep learning schemas, the consistency of IVs regression with deep networks is still an open problem.

These modifications make IV4Rec+ not only enjoy a number of merits from IVs, including the elegant approach to involving external search information for constructing IVs for recommendation and the regression for decomposing the treatment but also suitable for the recommendation task.

In the recommendation task, biases are ubiquitous, e.g., selection bias and position bias, while these biases are usually mixed and difficult to identify. In this article, in order to get rid of the difficulty of explicitly modeling multiple biases, IV4Rec+ focuses on improving the recommendation performance by using search data as IVs. IV4Rec+ explores the causal relationship between the search and recommendation tasks and enhances the recommendation model by reconstructing a unified treatment. Since IVs can be used to adjust for confounding effects, IV4Rec+ can be considered as a causal learning framework for recommendation using search data.

6 EXPERIMENTS

In this section, we conducted experiments to verify the effectiveness of the proposed IV4Rec+.

6.1 Experimental Settings

6.1.1 Datasets. IV4Rec+ requires user behaviors on both search engines and recommender systems. In the experiments, we used three datasets: two were collected from logs of the Kuaishou

Table 1. Statistics of Datasets

Dataset	User	Item	Query	Interaction
Kuaishou-small	12,000	3,053,966	162,624	4,001,613
Kuaishou-large	98,875	5,838,005	1,200,065	10,479,926
MIND	736,349	130,380	130,380	95,447,571

short-video app, and one was constructed on the basis of the publicly available MIND dataset [36]. Compared to the original article, we collected a new dataset, which contains substantially more user search and recommendation behaviors, to better evaluate our proposed methods.

Kuaishou-small Dataset: The Kuaishou-small dataset was created based on the activities of 12,000 randomly selected users when they elected to use both the search and recommendation services on an app named Kuaishou,[2] one of the largest short-video platforms in China, over a period of 7 days in May 2021. The historical behaviors in search and recommendation services of each user were collected. For each item and query in the dataset, the item embedding (64 dimensions), and query embedding (64 dimensions) were generated using existing pre-trained and ranking models from the platform.

We split the dataset into three subsets in chronological order, i.e., the first 5 days for training, the 6th day for validation, and the last day for testing. The mini-batch size is set to be 50.

Kuaishou-large Dataset: For conducting more convincing experiments, we also created another Kuaishou-large dataset which was collected from the same app in a period of 13 days in January 2022. The dataset contains 98,875 users who used both search and recommendation services during the period. This dataset is much larger than Kuaishou-small, for better evaluating the robustness and effectiveness of the models. The embeddings of items and queries are generated in similar ways as that of Kuaishou-small.

We split the dataset into three subsets in chronological order, i.e., the first 10 days for training, the 11th day for validation, and the last two days for testing. The mini-batch size is set to be 512.

MIND Dataset: To the best of our knowledge, there is no publicly available dataset that contains user activities in both search and recommendation. As a result, we enhanced the MIND[3] [36] dataset, a benchmark for news recommendation, by generating queries from the metadata. Specifically, motivated by the observation in [25], we generate one search query for each news article by concatenating the texts of its category, subcategory, and entities in the metadata. For a few articles whose entities are missing, NLTK[4] was used to extract entities from the titles. In this way, query-item pairs are constructed. For user search history, we construct user search queries by linking queries of each item interacted by users. To generate the query and item embeddings, we followed [36] and used BERT [9] to generate the item embeddings (768 dimensions) where the input is the concatenation of the title and abstract. Query embeddings (768 dimensions) are also generated by BERT with query strings as input. In order to accelerate the training, we applied a linear transformation after the embedding layer to reduce the data dimension from 768 to 64.[5]

We directly use the training and validation set provided by the MIND dataset.[3] Since MIND does not contain a test set with labels, the original training (and validation) set is used as the training (and test) set in the experiments. The mini-batch size is set to be 512.

Table 1 shows the basic statistics of these three datasets.

[2]https://www.kuaishou.com/en.
[3]https://msnews.github.io/.
[4]https://www.nltk.org/.
[5]Note that the setting is different from that of the original WWW 2022 article.

6.2 Baselines and Evaluation Metrics

The following sequential recommendation models are chosen as baselines in the experiments.

NRHUB [35]: NRHUB utilizes an attentive multi-view learning framework for news recommendation to aggregate heterogeneous behaviors of users, such as search queries, clicked items, and browsed items. In the experiments of the MIND dataset, it was adapted by removing the module using clicked items since only search queries were created. In the experiments of all datasets, the query encoder was removed since item embeddings were generated by pre-trained models.

DIN [56]: DIN applies an attention mechanism to mine user interests from historical behaviors w.r.t. a certain candidate item. The proposed local attention mechanism can capture diverse user interests.

SRGNN [39]: SRGNN models session sequences of user behavior as graph-structured data, and learns session embedding from session graphs by gated-GNN and applies an attention network to learn the global and current preferences.

For the input data of these models, NRHUB utilizes both recommendation history and search history. DIN, as well as SRGNN, only utilizes the recommendation history. Except for DIN on the Kuaishou-small dataset, we adapted it by adding search history into the input data, following the setting in the conference article.

We also compare IV4Rec+ to JSR [48] which jointly optimizes search and recommendation. JSR is a general joint learning framework that trains a separate search model and recommendation model by optimizing a joint loss. The search component of JSR was designed as a fully-connected feed-forward network, following the original article. The recommendation component was set as NRHUB, DIN, or SRGNN, leading to three versions of JSR: **JSR-NRHUB**, **JSR-DIN,** and **JSR-SRGNN**.

Besides, we compare IV4Rec+ to IV4Rec, which is proposed in the original WWW 2022 article [27]. Different from IV4Rec+, IV4Rec utilizes least square regression for IVs regression and constructs IVs for each item. We applied IV4Rec to the above-mentioned baselines, achieving three versions of IV4Rec, referred to as **IV4Rec-NRHUB**, **IV4Rec-DIN,** and **IV4Rec-SRGNN,** respectively.

Like IV4Rec, IV4Rec+ is also a model-agnostic and non-intrusive framework, which can take existing sequential recommendation models as the underlying models. In the experiments, IV4Rec+ is applied to the baselines mentioned above. Also note that IV4Rec+ has two variations according to the embeddings being decomposed, denoted as IV4Rec+(I) and IV4Rec+(UI), respectively. Therefore, IV4Rec+ has six variations, referred to as **IV4Rec+(UI)-NRHUB** and **IV4Rec+(I)-NRHUB**, **IV4Rec+(UI)-DIN** and **IV4Rec+(I)-DIN**, **IV4Rec+(UI)-SRGNN** and **IV4Rec+(I)-SRGNN**, respectively.

For performance evaluation metrics, we choose AUC, MRR, NDCG@5, and NDCG@10, which are widely adopted in many related works [35, 36]. Following the same way as the MIND[6] did, we calculated metrics on each impression list and reported the average results of all impressions, where an impression list contains click events and non-click events. We use the recommended lists for the Kuaishou-small and the Kuaishou-large datasets, which contain dozens of items for each refresh in the short-video app, to serve as impression lists. For the MIND dataset, impression lists are provided.

6.2.1 Implementation Details. The hyper-parameters of the neural networks mentioned in the experiments were optimized using grid search. The learning rate was selected from

[6] https://msnews.github.io/.

$\{1e\text{-}5, 5e\text{-}5, 1e\text{-}4, 5e\text{-}4, 1e\text{-}3\}$ and the dropout keep probability was selected from $\{0.8, 0.9, 1.0\}$. For the MLPs used in the proposed IV4Rec+, dropout was enabled, the activation function was ReLU and the depth of hidden layers was selected from $[1, 5]$ with step 1. The γ_1, γ_2, and γ_0 were searched from $[0.1, 0.9]$ with step 0.2 and $[0.01, 0.07]$ with step 0.03. The maximum length for the user interaction sequence was 50 for all datasets. As for the underlying models, we set the parameters as the optimal values reported in the original article. For making fair comparisons, each underlying model was set in an identical configuration in different deployed frameworks. For instance, JSR-NRHUB, IV4Rec-NRHUB, and IV4Rec+NRHUB use NRHUB as their underlying models, and these NRHUB models share identical settings. Adam [17] was used to conduct the optimization.

As described in Sections 4.2.1 and 4.3.1, we recall the queries that clicked an item to construct its IVs. In real-world data, the query-click data is very sparse and many items have not been clicked in search logs, as shown in Table 1. To address the data sparsity problem, we used cosine similarity to recall relevant queries for items. Cosine similarity of item and query embeddings was used to measure the strength of association where the embeddings were generated by a ranking model of the platform. Query-item pairs with high cosine similarity were used as complementary to the sparse query-click data.

6.3 Overall Performance Comparison

Table 2 presents the recommendation performance of all methods on the three datasets. From Table 2, we observed that:

- In all experiments, the two variants of IV4Rec+ (IV4Rec+(UI) and IV4Rec+(I)) boost the underlying models by a large margin. In most cases, these two variants significantly outperformed the strong baseline of IV4Rec (paired t-test at $p-$value < 0.01). The performance gain reveals the effectiveness of the proposed two strategies in improving any sequential recommendation models by using them as the underlying models. Also, the results verified the effectiveness of leveraging queries from the viewpoint of users and the multi-task training schema.
- Comparing the two variants IV4Rec+(UI) and IV4Rec+(I)), we find that they have comparable performances in most cases, and IV4Rec+(I) performed slightly better than IV4Rec+(UI). We postulate that the queries contain relatively immediate information needs of users. The user intentions and interests in queries evolve quickly. However, queries are highly relevant to the content of clicked items. The associations between queries and items are more stable. The phenomenon explains why IV4Rec+(I) achieves slightly better performances than IV4Rec+(UI).
- Compared to JSR, IV4Rec+ achieves much better performances on all of the three underlying models. We analyzed the reasons and found that JSR simply combines the search and recommendation models with a joint learning loss, neglecting the causal relations between them. IV4Rec+ utilizes different approaches to inject search queries into recommendation model and achieved the highest performance among all joint search and recommendation frameworks. The results verified the effects of using search queries as IVs for reconstructing the embeddings (as treatments) in recommendation.
- In terms of the implementations over NRHUB, our methods also lead to significant improvements. In light of the fact that NRHUB leverages search logs as additional features for user modeling, the improvements verify the conclusion that exploiting the causal relations between search and recommendation can further enhance underlying models, even though the search activities have been used as user features.

Table 2. Performance Comparisons between IV4Rec+ and the Baselines

Models	Kuaishou-large				Kuaishou-small				MIND			
	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10	AUC	MRR	nDCG@5	nDCG@10
DIN	0.6163	0.4989	0.3345	0.3495	0.6512	0.1833	0.4416	0.4743	0.6862	0.3308	0.3652	0.4285
JSR-DIN	0.6175	0.4934	0.3326	0.3505	0.6524	0.1838	0.4417	0.4755	0.6892	0.3325	0.3664	0.4305
IV4Rec-DIN	0.6223	0.5005	0.3426	0.3579	0.6561	0.1844	0.4432	0.4779	0.6913	0.3378	0.3731	0.4354
IV4Rec+(UI)-DIN	**0.6275***	0.5051*	0.3450	0.3610*	**0.6603***	0.1862	**0.4506***	0.4838*	**0.6953***	**0.3392***	**0.3761***	**0.4374***
IV4Rec+(I)-DIN	0.6269	**0.5092***	**0.3456**	**0.3619***	0.6599*	**0.1863**	0.4493*	**0.4839***	0.6944*	0.3387*	0.3761*	0.4381*
NRHUB	0.5996	0.4754	0.3137	0.3309	0.6455	0.1816	0.4347	0.4692	0.6707	0.3202	0.3509	0.4105
JSR-NRHUB	0.6021	0.4783	0.3157	0.3321	0.6488	0.1812	0.4326	0.4687	0.6711	0.3190	0.3508	0.4152
IV4Rec-NRHUB	0.6047	**0.4896**	0.3235	0.3416	0.6574	0.1837	0.4411	0.4774	0.6842	0.3306	0.3655	0.4282
IV4Rec+(UI)-NRHUB	**0.6137***	0.4854	**0.3267***	**0.3454***	0.6587	**0.1866***	0.4474*	0.4825*	0.6888*	0.3321*	0.3665	0.4296*
IV4Rec+(I)-NRHUB	0.6095*	0.4879	0.3257	0.3443*	**0.6593**	0.1862*	**0.4488***	**0.4827***	**0.6911***	**0.3347***	**0.3699***	**0.4326***
SRGNN	0.6047	0.4857	0.3231	0.3340	0.6312	0.1770	0.4162	0.4523	0.6403	0.3038	0.3296	0.3947
JSR-SRGNN	0.6054	0.4915	0.3253	0.3419	0.6318	0.1769	0.4181	0.4521	0.6398	0.3022	0.3289	0.3937
IV4Rec-SRGNN	0.6092	0.4831	0.3204	0.3382	0.6462	0.1804	0.4311	0.4663	0.6677	0.3136	0.3420	0.4082
IV4Rec+(UI)-SRGNN	0.6173*	0.4974*	0.3351*	0.3512*	0.6492*	0.1825	0.4386*	0.4712*	0.6699	0.3184*	0.3508*	0.4147*
IV4Rec+(I)-SRGNN	**0.6270***	**0.5009***	**0.3393***	**0.3571***	**0.6585***	**0.1858***	**0.4483***	**0.4801***	**0.6815***	**0.3316***	**0.3656***	**0.4281***

The bold font represents the best performance. The last two lines for each block are the two variants of IV4Rec+ with corresponding underlying model. For each underlying model, paired t-tests are conducted and "*" indicates the improvements against the best baselines are statistically significant (p-value < 0.01).

6.4 Empirical Analysis

In this section, we conducted more detailed experiments on the industrial Kuaishou-large dataset, for a better understanding of how and why IV4Rec+ improves the recommendation accuracy.

6.4.1 Feasibility of Using Search Queries as IVs. We conducted experiments to verify that search queries satisfy the relevance assumption of IVs, as mentioned in Section 5.1. As the search queries are utilized as IVs for clicked items and search query history as IVs for user historically interacted items, we examine the relevance from these two perspectives.

We use **distance correlation** (***dCor***) to measure the relevance between treatment variables and IVs. *dCor* is a measure of linear and non-linear association strength between multi-dimensional vectors, which ranges from 0 to 1, where $dCor(X, Y) = 0$ if X and Y are independent and $dCor(X, Y) = 1$ if X and Y are in equal linear sub-spaces. More details of *dCor* can be found in [29]. For queries and items, we calculate *dCor* of item embedding and its corresponding query embedding. For search query history and interacted item history, we first transform the query/item history sequences into vectors via average pooling, which applies element-wise average on query/item vectors. Then *dCor* of these vectors is calculated. For conducting better comparisons, we also calculate the *dCor* when queries are randomly sampled to compose the query and item pair (or search query history and interacted item history pair). From the results reported in Table 3,[7] we find that the collected pairs used in our methods are highly relevant, whose *dCor* is around 0.6. On the contrary, the sampled pairs are irrelevant with *dCor* around 0.1. The results confirm that using queries as IVs meet the relevance assumption.

6.4.2 Effects of Using Search Queries as IVs. We also conducted experiments to investigate how IV4Rec+ benefit from the IVs regression.

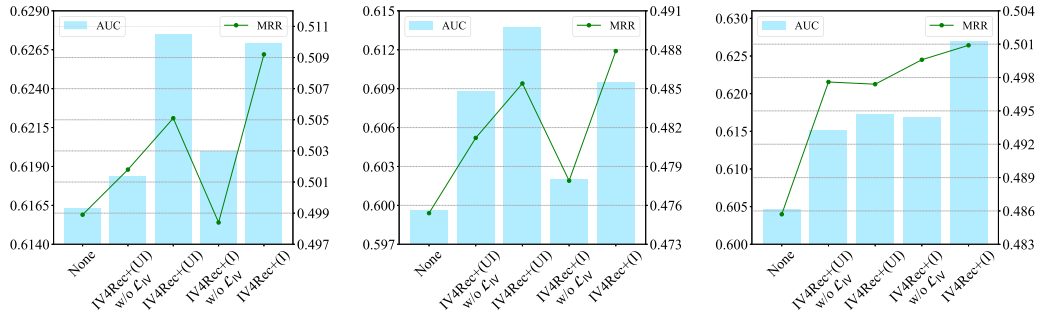
Impact of the IVs regression. First, we conducted experiments to show whether using search data as IVs is beneficial to the recommendation task. In light of the fact that most existing work focuses on leveraging search logs as external features for recommendation, we remove the IV regression loss $\mathcal{L}_{\text{IV}}$, which is key to our causal learning methods, to investigate the impact of the IVs regression. For two variants of IV4Rec+, the versions without $\mathcal{L}_{\text{IV}}$ are denoted as IV4Rec+(UI) w/o $\mathcal{L}_{\text{IV}}$

[7]Because *dCor* is computationally expensive, e.g., 100,000 samples require more than 5,120 GB of RAM in our case, we did the experiments by randomly sampling 10,000 pairs and reported average results after repeating 10 times.

Table 3. *dCor* w.r.t. Query and Item Pairs (Denoted as Item), and *dCor* w.r.t. user Search Queries and user Interacted Items Pairs (Denoted as User)

dCor	User	Item
collected pairs	0.5644	0.6102
randomly sampled pairs	0.1339	0.0873

Collected pairs are highly relevant in terms of *dCor* larger than 0.5, and randomly sampled pairs are irrelevant. The results verified that queries as IVs satisfy the relevance assumption. We refer the details of *dCor* to [29].

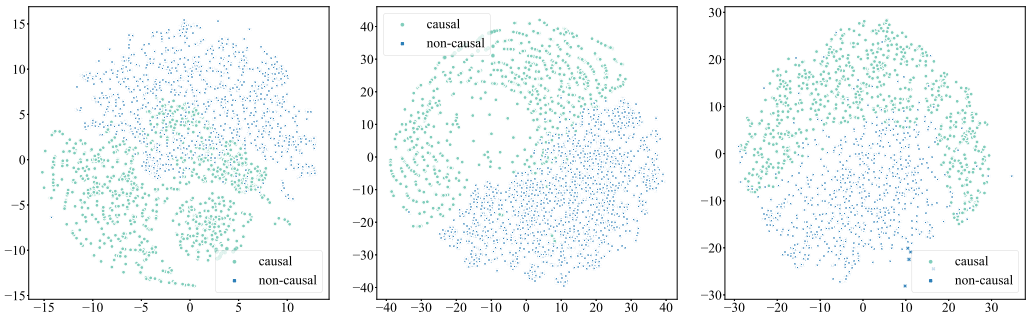


(a) DIN. (b) NRHUB. (c) SRGNN.

Fig. 6. Impact of IV regression loss for IV4Rec+(UI) and IV4Rec+(I) applied to three underlying models w.r.t AUC and MRR. "None" denotes the underlying models without using IV4Rec+ framework. The IV regression does help to improve the recommendation accuracy.

and IV4Rec+(I) w/o $\mathcal{L}_{\text{IV}}$ respectively. These two versions simply inject search queries into underlying models as additional features. Figure 6 illustrates the performances of IV4Rec+ and the variations without $\mathcal{L}_{\text{IV}}$, applied to DIN, NRHUB, and SRGNN. In most cases, we can find that AUC and MRR drop a lot after removing the IV regression loss $\mathcal{L}_{\text{IV}}$, for both the underlying models of DIN, NRHUB, and SRGNN. For instance, for DIN, AUC drops 0.06+ points for IV4Rec+(UI) or IV4Rec+(I). The phenomenon verifies the necessity and effectiveness of the IVs regression in IV4Rec+.

IV4Rec+ has the ability to disentangle causal and non-causal parts of treatment vectors, i.e., decomposing the treatment $\mathcal{T}_{u,i}$ into $\widehat{\mathcal{T}}_{u,i}$ and $\widetilde{\mathcal{T}}_{u,i}$. In causal graph Figure 2(b), the fitted part is independent of confounders, and the residual part is relevant to confounders. We conduct experiments to verify whether IV4Rec+ achieves disentanglement after treatment decomposition. Towards this end, we visualize the decomposed two parts of item and user embeddings learned by IV4Rec+ using t-SNE [30]. Figure 7 shows learned item embeddings from IV4Rec+(I) and learned user/item embeddings from IV4Rec+(UI), respectively. The fitted part is represented by *dots* and the residual part is represented by *crosses*. We see that with the designed IVs regression, there is a clear separation between the two sets of embeddings. We also notice that there is a small overlap between two sets of embeddings, indicating the difficult cases (users/items) to disentangle. The reason may be that the relevance between these queries-items/users pairs is not significant enough to leverage queries as valid IVs. Overall, we conclude that IV4Rec+ successfully captures the different parts of treatment vectors, which can be utilized in different ways to enhance recommendation models.



(a) Item embeddings learned by IV4Rec+(I).

(b) Item embeddings learned by IV4Rec+(UI).

(c) User embeddings learned by IV4Rec+(UI).

Fig. 7. Visualization of the learned causal and non-causal embeddings of IV4Rec+. Causal parts are represented by dots and non-causal parts are represented by crosses. Causal and non-causal parts are disentangled clearly by IV4Rec+.

Robustness of IV4Rec+. To show the robustness of the proposed IV4Rec+, we conducted analysis on its performance w.r.t. the number of search logs.

On the test set of Kuaishou-large dataset, we group users into three parts by their search activities. Firstly, we calculate the ratio of the number of days in user history that contain search activities to the number of days in the whole user history. Formally, the ratio is calculated as $\frac{\sum_{j \in \mathcal{D}_{\text{his}}} \mathbb{I}_j}{\| \mathcal{D}_{\text{his}} \|}$ where $\mathcal{D}_{\text{his}}$ is the set of days in user history and $\mathbb{I}_j$ is an indicator, i.e., $\mathbb{I}_j = 1$ if day j contains search activities, $\mathbb{I}_j = 0$ otherwise. Then users with ratio in $[0, \frac{1}{3})$ (3,272 low active users), $[\frac{1}{3}, \frac{2}{3})$ (2,135 medium active users), and $[\frac{2}{3}, 1]$ (1,271 high active users) are separated into three groups and denoted as low, medium, and high, respectively.

We conducted experiments on the two aforementioned variants of IV4Rec+ and a new variant which removes the IVs part for the candidate item in IV4Rec+(UI), i.e., only leverages search history as IVs for users, denoted as IV4Rec+(U). The IV4Rec+(U) can be seen as IV4Rec+(UI) without using IVs part for the candidate item. The IV4Rec+(U) removes the candidate item part in Equations (2), (4), (5), and (6), e.g., $\mathbf{z}_i$ in Equation (2). We created the IV4Rec+(U) to explore the effect of the number of search logs on IV4Rec+(UI). We did not do the same things on IV4Rec+(I) because it collects queries related to items to construct IVs so that the number of search logs have no direct effect on it. We implement these three frameworks over DIN and SRGNN. NRHUB is omitted here because its user modeling module also uses search logs, leading to distortion in analysis on the robustness of IV4Rec+. The models are evaluated on three subsets of the test set as well as the overall test set. We show the performance in terms of AUC and adopt the RelaImpr [56] to measure the relative improvements over models, which is defined as

$$\text{RelaImpr} = \left(\frac{\text{AUC(measured model)} - 0.5}{\text{AUC(base model)} - 0.5} - 1 \right) \times 100\%.$$

From Table 4, we can find that in most cases IV4Rec+(I) and IV4Rec+(UI) make consistent improvements over corresponding underlying models, on all of the user groups. The results indicate that IV4Rec+ is robust in terms of improving the experience of all types of users. We analyzed the reasons. For IV4Rec+(I), note that the collected queries (IVs) corresponding to items are not affected by the user's search activity level. For IV4Rec+(UI), though the queries (IVs) are sparse for low active users, the queries for the items are still helpful to the overall performances. Therefore, IV4Rec+(UI) is more robust than IV4Rec+(U). Moreover, the performance comparisons reveal

Table 4. Performance Comparison of DIN and SRGNN with Different IV4Rec+ Variants in Terms of AUC

Model	Framework	Kuaishou-large							
		low		medium		high		overall	
DIN	-	0.6140	-	0.6211	-	0.6142	-	0.6163	-
	IV4Rec+(U)	0.6199	+5.17%	0.6287	+6.27%	0.6230	+7.70%	0.6233	+6.01%
	IV4Rec+(I)	0.6251	+9.73%	0.6312	+8.34%	0.6245	+9.01%	0.6269	+9.11%
	IV4Rec+(UI)	0.6266	+11.05%	0.6312	+8.34%	0.6222	+8.49%	0.6275	+9.63%
SRGNN	-	0.6051	-	0.6077	-	0.5959	-	0.6042	-
	IV4Rec+(U)	0.6140	+8.46%	0.6141	+5.94%	0.6140	+18.87%	0.6146	+9.98%
	IV4Rec+(I)	0.6280	+21.78%	0.6316	+22.19%	0.6164	+21.37%	0.6270	+21.88%
	IV4Rec+(UI)	0.6179	+12.17%	0.6213	+12.62%	0.6089	+13.55%	0.6173	+12.57%

Relative improvements over DIN or SRGNN are in percentages. Low, medium, and high denote users with different search activity in the test set separately. Overall denotes all users in the test set. RelaImpr is adopted to measure AUC improvements over underlying models, i.e., DIN and SRGNN.

Table 5. The Performance of IV4Rec+ over SASREC and GRU4REC

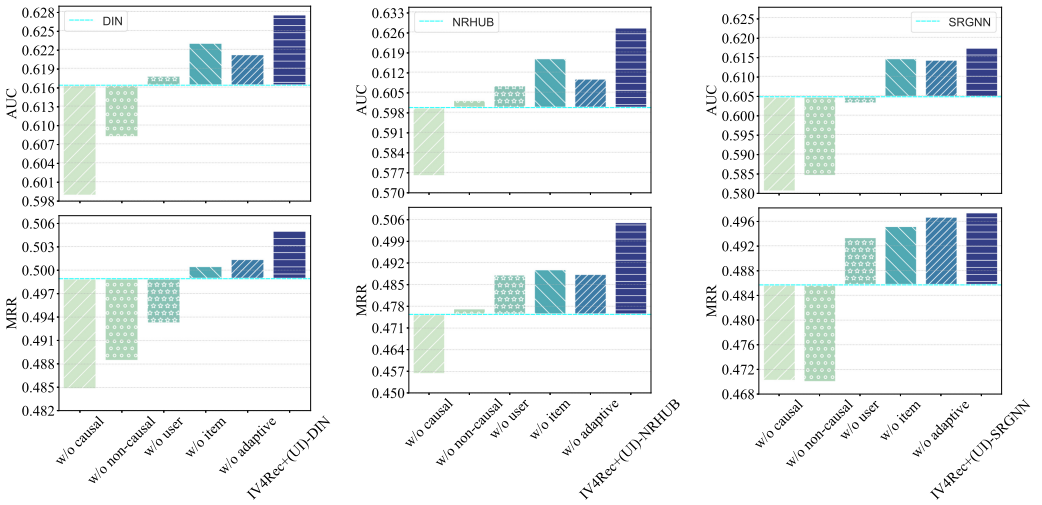
Model	Kuaishou-large			
	AUC	MRR	nDCG@5	nDCG@10
SASREC	0.6134	0.4954	0.3297	0.3475
IV4Rec+(UI)-SASREC	0.6175	0.5005	0.3375	0.3528
IV4Rec+(I)-SASREC	0.6236	0.5051	0.3424	0.3588
GRU4REC	0.6092	0.4733	0.3163	0.3361
IV4Rec+(UI)-GRU4REC	0.6121	0.4867	0.3284	0.3445
IV4Rec+(I)-GRU4REC	0.6204	0.5020	0.3378	0.3539

that the IV4Rec+(U) is vulnerable to user search activity levels. We observed that IV4Rec+(U) better boosts DIN and SRGNN with more search queries. Contrary to IV4Rec+(U), IV4Rec+(UI) has obtained consistent improvements with different search activity level users, indicating that the candidate item part helps the framework to get stable performance. Compared to IV4Rec+(I), IV4Rec+(UI) is less robust due to the impacts of the user search activity levels. This also explains why IV4Rec+(I) is slightly better than IV4Rec+(UI) in Table 2.

6.4.3 Generality of IV4Rec+. We conducted experiments to explore whether the IV4Rec+ can be applied to typical sequential recommendation methods.

From the results in Table 2, we observed that the IV4Rec+(UI) and IV4Rec+(I) can enhance sequential models with different mechanisms, i.e., DIN with the target attention mechanism, NRHUB with the additive attention mechanism and SRGNN with the gated GNN and soft attention mechanisms. To further explore whether the frameworks can boost any typical sequential models, we apply the IV4Rec+ over two representative models, SASREC [16] using the self-attention mechanism and GRU4REC [13] using the gated recurrent unit. The Table 5 represents the results of the experiments.

From the Table 5, we found that the IV4Rec+(I) can boost the given types of sequential models consistently. IV4Rec+(UI) gets minor improvements than IV4Rec+(I) over SASREC and GRU4REC. Through observations from Tables 2 and 5, we conclude that the IV4Rec+(I) can obtain consistent improvements over most representative sequential models. And the IV4Rec+(UI) can perform well over most sequential models, but with minor improvements on several models than IV4Rec+(I). We attribute the minor improvements to IV4Rec+(UI)'s IVs construction for the user embedding. IV4Rec+(UI) calculates the user search embedding through additional attention networks. Thus



(a) DIN. (b) NRHUB. (c) SRGNN.

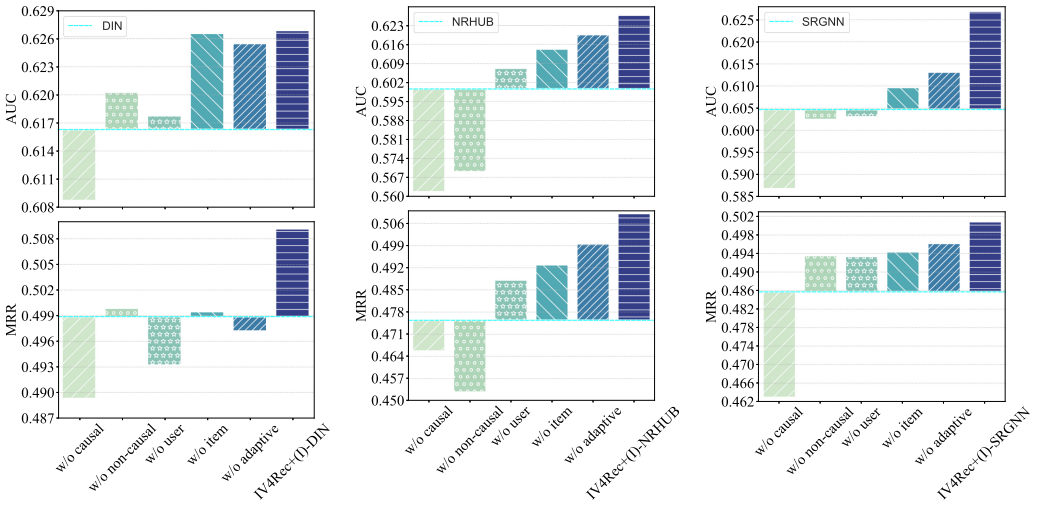
Fig. 8. Ablation study of IV4Rec+(UI) over three underlying models w.r.t AUC and MRR. "w/o" indicates that the corresponding component is removed while the rest components are kept. Horizontal lines denote performance of three underlying models, respectively.

the additional attention networks may be not expressive enough to aggregate user search interest w.r.t some kinds of sequential models.

6.4.4 Ablation Study. IV4Rec+ consists of several key operations, including using both causal and non-causal parts of the treatments, constructing IVs for users and items, the adaptive fusion of the causal and non-causal parts, and so on. To figure out the effects of each operation, we conducted several ablation experiments for IV4Rec+(UI) and IV4Rec+(I) over three underlying models. The results are shown in Figures 8 and 9, respectively. The results of the underlying models are also provided in the figures for better comparison. From the results, we observe that removing any component results in a performance decline (compared to IV4Rec+(UI) and IV4Rec+(I)), implying that each of the operations has contributed to improving the AUC and MRR. Also, the importance of each component varies from different underlying models, which is attributed to different model architectures. Next, we give a detailed discussion about each component:

Importance of utilizing both causal and non-causal parts. To capture different relations between user-item pairs and feedback, IV4Rec+ utilizes both causal and non-causal parts. We alternately remove these two parts to verify their effectiveness. In the experiments, we do not directly remove the fitted part $\widehat{\mathcal{T}}_{u,i}$ or the residuals $\widetilde{\mathcal{T}}_{u,i}$ from the models. The reason is that $\mathcal{T}^{\text{re}}_{u,i}$ would only consist of $\widehat{\mathcal{T}}_{u,i}$, and treatment modules of the underlying models can not be updated with gradients if $\widetilde{\mathcal{T}}_{u,i}$ were removed. Also, if treatments are only learned by non-causal signals without $\widehat{\mathcal{T}}_{u,i}$, the optimization algorithms are hard to converge. To address these issues, we used the "detach" trick[8] in the experiments. In order to remove the effects of $\widetilde{\mathcal{T}}_{u,i}$, $\mathcal{T}^{\text{re}}_{u,i}$ is calculated as $\widehat{\mathcal{T}}_{u,i}$ + $\widetilde{\mathcal{T}}_{u,i}$ - $\widetilde{\mathcal{T}}_{u,i}$.datach() where "detach()" denotes the function that cuts off the gradients in PyTorch. Similar operations are done to remove the effects of $\widehat{\mathcal{T}}_{u,i}$.

[8]We refer to [15] for more details on this trick.



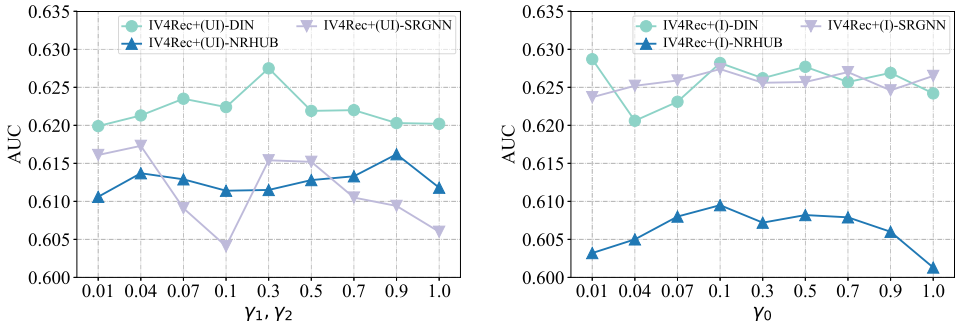
(a) DIN. (b) NRHUB. (c) SRGNN.

Fig. 9. Ablation study of IV4Rec+(I) over three underlying models w.r.t AUC and MRR.

The results are reported in the leftmost two bars (denoted as "w/o causal" and "w/o non-causal") in the sub-figures in Figures 8 and 9. After removing the causal or non-causal part, the performances drop drastically. In most cases, they are significantly worse than the underlying models. The phenomenon implies that the causal and non-causal parts do play important roles in the preference prediction. Especially, removing the causal part brings a more drastic decrease. The causal part represents the user preference for the target item, which is the main reason for click behaviors. That is the reason why the causal part contributes so much to prediction accuracy. Also, we observe that removing the causal part or the non-causal part causes difficult convergence of the model. That is because discarding any of them would lead to new treatment vectors in different vector spaces from the original model. That also explains why we use the linear combination to re-construct the treatments.

Necessity of constructing IVs for both users and items. In IV4Rec+(UI) or IV4Rec+(I), we collect search query history or corresponding queries as IVs for users' interacted items. We also collect the clicked queries as IVs for candidate items. To confirm the respective effects of the IVs for users and items, we alternatively strip off the two parts. Note that without IVs for users, IV4Rec+(UI) and IV4Rec+(I) degenerate to the same structure because the differences between them are the IVs construction for users. From results reported in two middle bars (denoted as "w/o user" and "w/o item") of the sub-figures in Figures 8 and 9, we find that removing either IVs for users or IVs for items leads to performance decline. The results indicate that using queries as IVs from the viewpoint of either users or items is indispensable. Besides, the performance of models without IVs for users decreases more than those without IVs for items. For instance, IV4Rec+(I) over DIN loses 3.14% in MRR and 1.45% in AUC without IVs for users and loses 1.90% in MRR and 0.04% in AUC without IVs for items. We postulate that the user logs usually contain noisy interactions affected by confounders. Thus our frameworks gain more benefits on the user part. The two types of IVs contribute to the final preference prediction from different perspectives.

Adaptive combination of causal and non-causal parts. After decomposing treatments, IV4Rec+ aggregates the two parts with an MLP by learning adaptive weights according to the fitted part and the original treatment. To verify the effectiveness of the mechanism, we disable the



(a) IV4Rec+(UI). (b) IV4Rec+(I).

Fig. 10. Impact of hyper-parameter γ_1, γ_2 for IV4Rec+(UI) or γ_0 for IV4Rec+(I) in terms of AUC. To see an overall trend, we set $\gamma_1 = \gamma_2$ in these experiments.

adaptive weights to confirm the effectiveness of this module. Specifically, simple concatenation is used to combine the two parts together, which means these two parts contribute to the final recommendation with fixed weights. Results are shown in the bars denoted as "w/o adaptive" in the sub-figures of Figures 8 and 9. The results indicate that the proposed IV4Rec+ outperforms the variation where the adaptive combination is removed. These results verify the advantages of using adaptive weights to reconstruct the causal and non-causal parts of the treatments.

6.4.5 Effects of Hyper-parameters. IV4Rec+ depends on a multi-task learning schema containing the causal learning task and recommendation model learning task. Several hyper-parameters are used to balance these objectives in the loss function, including γ_0 in IV4Rec+(I) and γ_1, γ_2 in IV4Rec+(UI). To investigate the effectiveness of these hyper-parameters, we conduct parameter sensitivity analysis.

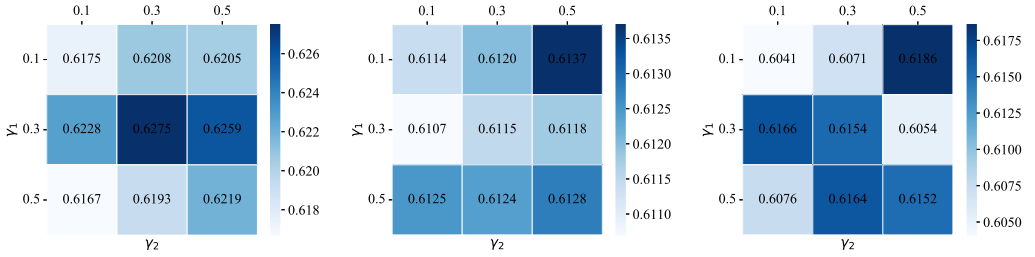
As formulated in Section 4.4, γ_0 in IV4Rec+(I) and γ_1, γ_2 in IV4Rec+(UI) are introduced to balance the impact of different objectives and re-scale the IV loss in joint learning. To investigate the impacts of these hyper-parameters, we conduct experiments by applying IV4Rec+(UI) and IV4Rec+(I) over three underlying models with varying γ_0 or γ_1, γ_2. In particular, we vary these parameters in the ranges of $[0.01, 0.07]$ with step 0.03, and $[0.1, 0.9]$ with step 0.2. As for comparisons, we also test the model performances when fixing γ_0 and γ_1, γ_2 to 1.0. For IV4Rec+(UI), we set $\gamma_1 = \gamma_2$ in the experiments.

According to the results shown in Figure 10, we found that the performances of IV4Rec+(UI) and IV4Rec+(I) do not vary dramatically with different γ values, indicating that the models are robust and not very sensitive to the setting of the hyper-parameters. In most cases, the performance curves drop when γ's are close to 0 and 1.0, indicating the necessity of balancing these losses.

We further study the impacts of γ_1 and γ_2 in IV4Rec+(UI). Specifically, we set γ_1 and γ_2 in $\{0.1, 0.3, 0.5\}$ and visualize the results in the heat map Figure 11. The colors and numbers in the blocks indicate the AUC scores. From the results, we found that varying γ_2 when fixing γ_1 has a weaker impact on performance than that of varying γ_1 when fixing γ_2. We analyzed the reasons and found that the IVs for users contribute more than the IVs for items, as shown in Section 6.4.4.

7 CONCLUSIONS AND FUTURE WORK

In this article, we proposed a model agnostic IV-based causal learning framework to improve recommendation using search data, called IV4Rec+. The proposed framework first decomposes the recommendation embeddings into the causal association part and the non-causal association part



(a) IV4Rec+(UI)-DIN. (b) IV4Rec+(UI)-NRHUB. (c) IV4Rec+(UI)-SRGNN.

Fig. 11. Visualization of performance of different γ_1 and γ_2 for IV4Rec+(UI) w.r.t AUC.

with IV regression, and then combines two parts as a reconstructed representation for the next step interaction prediction. An end-to-end multi-task learning schema is developed to learn the model parameters. In particular, two strategies are proposed to inject the related search queries into the training of recommendation models in a causal learning manner: IV4Rec+(UI) and IV4Rec+(I). These strategies utilize different means of collecting queries to serve as IVs for user history. Both strategies are model agnostic. They can be easily applied over many existing recommendation models, which makes IV4Rec+ more flexible. Experiments on two industrial datasets and a public benchmark demonstrate the effectiveness of IV4Rec+ in recommendation.

As for future work, this work points to new research possibilities. Specifically, leveraging IVs to boost recommendation models is not limited to search data. Commercial streaming media platforms usually deploy advertisement content and media content in the same app. Moreover, online platforms usually provide comments section. Thus users' comment logs and advertisement logs have the potential to serve as IVs. We will explore how to incorporate various user behaviors into our framework.

ACKNOWLEDGMENTS

We would like to thank Xueran Han and Yue Yin for their contributions to the conference version of the article.

REFERENCES

[1] A. Belloni, D. Chen, V. Chernozhukov, and C. Hansen. 2012. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica* 80, 6 (2012), 2369–2429.

[2] Aman Agarwal, Ivan Zaitsev, Xuanhui Wang, Cheng Li, Marc Najork, and Thorsten Joachims. 2019. Estimating position bias without intrusive interventions. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.), 474–482.

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

[4] Mehmet Caner and Bruce E. Hansen. 2004. Instrumental variable estimation of a threshold model. *Econometric Theory* 20, 5 (2004), 813–843.

[5] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten De Rijke. 2019. Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems* 37, 4, (2019), 30 pages. DOI: https://doi.org/10.1145/3343117

[6] Yutian Chen, Liyuan Xu, Caglar Gulcehre, Tom Le Paine, Arthur Gretton, Nando de Freitas, and Arnaud Doucet. 2022. On instrumental variable regression for deep offline policy evaluation. *Journal of Machine Learning Research* 23, 302 (2022), 1–40. http://jmlr.org/papers/v23/21-0614.html.

[7] V. Chernozhukov, G. W. Imbens, and W. K. Newey. 2007. Instrumental variable estimation of nonseparable models. *Journal of Econometrics* 139, 1 (2007), 4–14.

[8] W. Bruce Croft, Donald Metzler, and Trevor Strohman. 2009. *Search Engines: Information Retrieval in Practice*. Pearson Education.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Jill Burstein, Christy Doran, and Thamar Solorio (Eds.), Association for Computational Linguistics, 4171–4186.

[10] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems* 39, 1 (2020), 42 pages. DOI: https://doi.org/10.1145/3426723

[11] Hector Garcia-Molina, Georgia Koutrika, and Aditya Parameswaran. 2011. Information seeking: Convergence of search, recommendations, and advertising. *Communications of the ACM* 54, 11 (2011), 121–130.

[12] Jason Hartford, Greg Lewis, Kevin Leyton-Brown, and Matt Taddy. 2017. Deep IV: A flexible approach for counterfactual prediction. In *Proceedings of the International Conference on Machine Learning*. PMLR, 1414–1423.

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations (ICLR'16, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings)*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939

[14] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds. 2021. Debiasing career recommendations with neural fair collaborative filtering. In *Proceedings of the Web Conference 2021*. 3779–3790.

[15] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *Proceedings of the 5th International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=rkE3y85ee.

[16] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *Proceedings of the 2018 IEEE International Conference on Data Mining*. IEEE, 197–206.

[17] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*. Yoshua Bengio and Yann LeCun (Eds.).

[18] Joshua D. Angrist and Jörn-Steffen Pischke. 2008. *Mostly Harmless Econometrics: An Empiricist's Companion*. Princeton University Press, 83–160.

[19] Anagha Kulkarni and Jamie Callan. 2015. Selective search: Efficient and effective search of large textual collections. *ACM Transactions on Information Systems* 33, 4 (2015), 33 pages. DOI: https://doi.org/10.1145/2738035

[20] Dugang Liu, Pengxiang Cheng, Hong Zhu, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2021. *Mitigating Confounding Bias in Recommendation via Information Bottleneck*. Association for Computing Machinery, 351–360. DOI: https://doi.org/10.1145/3460231.3474263

[21] Shiyang Lu, Tao Mei, Jingdong Wang, Jian Zhang, Zhiyong Wang, and Shipeng Li. 2014. Browse-to-search: Interactive exploratory search with visual entities. *ACM Transactions on Information Systems* 32, 4 (2014), 27 pages. DOI: https://doi.org/10.1145/2630420

[22] Robert E. McCulloch, Rodney A. Sparapani, Brent R. Logan, and Purushottam W. Laud. 2021. Causal inference with the instrumental variable approach and Bayesian nonparametric machine learning. arXiv preprint arXiv:2102.01199 (2021).

[23] Judea Pearl. 2009. *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

[24] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2011. *Recommender Systems Handbook*. Springer.

[25] Jennifer Rowley. 2000. Product search in e-sProduct search in e-shopping: A review and research propositionshopping: A review and research propositions. *Journal of Consumer Marketing* 17, 1 (2000), 20–35.

[26] Devashish Shankar, Sujay Narumanchi, H. A. Ananya, Pramod Kompalli, and Krishnendu Chaudhury. 2017. Deeplearning based large scale visual recommendation and search for e-commerce. arXiv preprintarXiv:1703.02344 (2017).

[27] Zihua Si, Xueran Han, Xiao Zhang, Jun Xu, Yue Yin, Yang Song, and Ji-Rong Wen. 2022. A model-agnostic causal learning framework for recommendation using search data. In *Proceedings of the ACM Web Conference 2022*. Association for Computing Machinery, 224–233. DOI: https://doi.org/10.1145/3485447.3511951

[28] Rahul Singh, Maneesh Sahani, and Arthur Gretton. 2019. *Kernel Instrumental Variable Regression*. Curran Associates Inc., Red Hook.

[29] Gábor J. Székely, Maria L. Rizzo, and Nail K. Bakirov. 2007. Measuring and testing dependence by correlation of distances. *The Annals of Statistics* 35, 6 (2007), 2769–2794.

[30] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605. http://jmlr.org/papers/v9/vandermaaten08a.html.

[31] Arun Venkatraman, Wen Sun, Martial Hebert, J. Andrew Bagnell, and Byron Boots. 2016. Online instrumental variable regression with applications to online linear system identification. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

[32] Jian Wang, Yi Zhang, and Tao Chen. 2012. Unified recommendation and search in e-commerce. In *Proceedings of the Information Retrieval Technology*. Yuexian Hou, Jian-Yun Nie, Le Sun, Bo Wang, and Peng Zhang (Eds.), Springer Berlin, 296–305.

[33] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue *(SIGIR'21)*. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 1288–1297. DOI: https://doi.org/10.1145/3404835.3462962

[34] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. *Counterfactual Data-augmented Sequential Recommendation*. Association for Computing Machinery, 347–356. DOI: https://doi.org/10.1145/3404835.3462855

[35] Chuhan Wu, Fangzhao Wu, Mingxiao An, Tao Qi, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with heterogeneous user behavior. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 4874–4883.

[36] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[37] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning fair representations for recommendation: A graph-based perspective. In *Proceedings of the Web Conference 2021*. 2198–2208.

[38] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. Pascal Van Hentenryck and Zhi-Hua Zhou (Eds.), 346–353. DOI: https://doi.org/10.1609/aaai.v33i01.3301346

[39] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 01 (2019), 346–353. DOI: https://doi.org/10.1609/aaai.v33i01.3301346

[40] Tao Wu, Ellie Ka-In Chio, Heng-Tze Cheng, Yu Du, Steffen Rendle, Dima Kuzmin, Ritesh Agarwal, Li Zhang, John Anderson, Sarvjeet Singh, Tushar Chandra, Ed H. Chi, Wen Li, Ankit Kumar, Xiang Ma, Alex Soares, Nitin Jindal, and Pei Cao. 2020. Zero-shot heterogeneous transfer learning from recommender systems to cold-start search retrieval. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. 2821–2828.

[41] Jun Xu, Xiangnan He, and Hang Li. 2018. Deep learning for matching in search and recommendation. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1365–1368.

[42] Liyuan Xu, Yutian Chen, Siddarth Srinivasan, Nando de Freitas, Arnaud Doucet, and Arthur Gretton. 2021. Learning deep features in instrumental variable regression. In *Proceedings of the International Conference on Learning Representations*.

[43] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems* 37, 3 (2019), 1–25.

[44] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. *Top-N Recommendation with Counterfactual User Preference Simulation*. Association for Computing Machinery, 2342–2351. DOI: https://doi.org/10.1145/3459637.3482305

[45] Jing Yao, Zhicheng Dou, Ruobing Xie, Yanxiong Lu, Zhiping Wang, and Ji-Rong Wen. 2021. USER: A unified information search and recommendation model based on integrated behavior sequence. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM'21, Virtual Event, Queensland, Australia)*, Association for Computing Machinery, New York, NY, 2373–2382. https://doi.org/10.1145/3459637.3482489

[46] Jiawei Yao, Jiajun Yao, Rui Yang, and Zhenyu Chen. 2012. Product recommendation based on search keywords. In *2012 9th Web Information Systems and Applications Conference*. IEEE, 67–70.

[47] Junkun Yuan, Anpeng Wu, Kun Kuang, Bo Li, Runze Wu, Fei Wu, and Lanfen Lin. 2022. Auto IV: Counterfactual prediction via automatic instrumental variable decomposition. *ACM Transactions on Knowledge Discovery from Data* 16, 4 (2022), 1–20.

[48] Hamed Zamani and W. Bruce Croft. 2018. Joint modeling and optimization of search and recommendation. In *Proceedings of the 1st Biennial Conference on Design of Experimental Search and Information Retrieval Systems*. Omar Alonso and Gianmaria Silvello (Eds.), CEUR-WS.org, 36–41.

[49] Hamed Zamani and W. Bruce Croft. 2020. Learning a joint search and recommendation model from user-item interactions. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 717–725.

[50] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the SIGIR'21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (Eds.), ACM, 367–377. DOI: https://doi.org/10.1145/3404835.3462908

[51] Xiao Zhang, Sunhao Dai, Jun Xu, Zhenhua Dong, Quanyu Dai, and Ji-Rong Wen. 2022. Counteracting user attention bias in music streaming recommendation via reward modification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2504–2514.

[52] Xiao Zhang, Haonan Jia, Hanjing Su, Wenhan Wang, Jun Xu, and Ji-Rong Wen. 2021. Counterfactual reward modification for streaming recommendation with delayed feedback. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 41–50.

[53] 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*. Association for Computing Machinery, 11–20.

[54] Zihao Zhao, Jiawei Chen, Sheng Zhou, Xiangnan He, Xuezhi Cao, Fuzheng Zhang, and Wei Wu. 2021. Popularity bias is not always evil: Disentangling benign and harmful bias for recommendation. *IEEE Transactionson Knowledgeand Data Engineering* (2022), 1–13. https://doi.org/10.1109/TKDE.2022.3218994

[55] 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.

[56] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1059–1068.

Received 11 June 2022; revised 15 November 2022; accepted 16 January 2023