CSRec: Rethinking Sequential Recommendation from A Causal Perspective.

Xiaoyu Liu* **Jiaxin Yuan*** **Yuhang Zhou** **Jingling Li** **Furong Huang** **Wei Ai**
University of Maryland, College Park
{xliu1231, jyuan98, tonyzhou, jingling, furongh, aiwei}@umd.edu
* contributes equally to the project

Abstract

The essence of sequential recommender systems (RecSys) lies in understanding how users make decisions. Most existing approaches frame the task as a sequential prediction based on users' historical purchase records. While effective in capturing users' natural preferences, this formulation falls short in accurately modeling actual recommendation scenarios, particularly in accounting for how unsuccessful recommendations influence future purchases. Furthermore, the impact of the RecSys itself on users' decisions has not been appropriately isolated and quantitatively analyzed. To address these challenges, we propose a novel formulation of sequential recommendation, termed Causal Sequential Recommendation (CSRec). Instead of predicting the next item in the sequence, CSRec aims to predict the probability of a recommended item's acceptance within a sequential context and backtrack how current decisions are made. Critically, CSRec facilitates the isolation of various factors that affect users' final decisions, especially the influence of the recommender system itself, thereby opening new avenues for the design of recommender systems. CSRec can be seamlessly integrated into existing methodologies. Experimental evaluations on both synthetic and real-world datasets demonstrate that the proposed implementation significantly improves upon state-of-the-art baselines. github link here.

1 Introduction

In the contemporary data-driven landscape, the significance of building a recommender system (recsys) cannot be overstated. These systems form the foundation for delivering personalized user experiences, improving customer satisfaction, and optimizing decision-making processes across a wide range of industries.

The fundamental objective of recommender systems is to understand how users make decisions and to provide support in this process. Most existing works, however, equate this objective with recommending the item that users are likely to prefer the most. Consequently, these approaches focus on estimating users' inherent preferences based on observational data, such as purchasing records, click data, and item reviews [18], and develop various methods to mitigate biases, such as popularity bias and exposure bias [15; 8]. Later, the study of sequential recommendation recognizes that users' preferences are also affected by temporal factors, taking into consideration that preferences may evolve over time or that previous purchase decisions can influence future choices. Consequently, recommendation tasks are formulated to predict the most likely next purchased item based on previous purchases, offering a more personalized recommendation [31; 30; 27].

However, recovering users' preferences is not fully equivalent to understanding the decision-making process. Despite their notable success, such formulations tend to oversimplify the complexities of real-world recommendation scenarios and fall short in precisely characterizing the recommendation

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

process. This oversimplification, to some extent, limits the potential for developing truly effective recommender systems.

Recommendation differs from users' natural selection processes in that the recsys controls how items are exposed to users, effectively serving as an *intervention* in their purchase decisions. In the formulation of predicting the next possible purchase, predictions are typically based on the user's previous acceptance of a series of items. However, it is entirely plausible that a user may dislike the system's recommendations or show no interest in the suggested items, leading to sequences that include both acceptance and rejection of items. Unfortunately, these possible sequences are excluded from current formulations[37; 27; 15] due to the fundamental difference in the training goal.

In addition, an increasing number of recent works have recognized the interventional nature of recommender systems and have attempted to model the system's influence on users[9; 12]. For instance, [9] highlighted the phenomenon of echo chambers, where users receive increasingly homogenized recommendations. Similarly, [12] sought to address the issue of amplified preference loops, where user preferences in observational data are repeatedly reinforced, ultimately undermining the personalized recommendation experience. The emergence of these problems underscores the need for an analytic framework capable of measuring the system's influence on users in a differentiable manner. While specific solutions have been proposed, they often overlook the fact that users' decisions are collaboratively influenced by multiple factors, not just the system itself. This reveals a gap in the existing methods, which lack the ability to isolate the system's influence from other factors, such as users' inherent preferences or previous purchase decisions.

In this paper, we introduce a novel formulation called CSRec (**C**ausal **S**equential **R**ecommendation), which can be integrated into existing recommender systems that preserves the underlying causal structure in sequential recommendations, thereby overcoming existing limitations. Given a causal graph, the causal inference perspective offers an explicit and differentiable framework to isolate and quantify the influence of each variable on a user's decision. While the underlying causal graph may differ across contexts, the same procedure can be consistently applied, making it a flexible and broadly applicable framework. Moreover, we introduce a method to quantitatively analyze the influence of recommender systems on users through treatment effect estimation, defined as the average difference in a user's decisions with and without the recommendation. This formulation opens up new possibilities for developing personalized recommender systems. In Section E, we present several examples demonstrating how this approach can be used as an evaluation metric to assess whether users are building trust in the system. Additionally, it offers a framework for developing strategies to balance the exploration-exploitation trade-off when navigating users' preferences in unknown domains.

We conduct extensive experiments on both synthetic and real-world dataset and empirically show that this simple adjustment of existing baselines generally improves the performance and outperforms SOTA baselines.

2 Sequential Recommendation Represented in Causal Graph

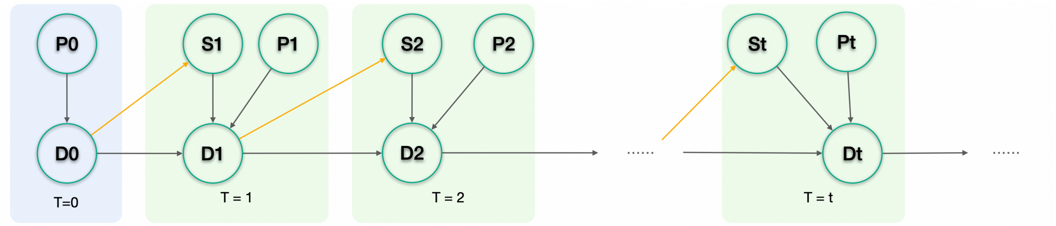

Figure 1: **Causal Graph for Sequential Recommendation.**

In this section, we introduce the causal graph for the sequential recommendation as shown in Figure 1.

In this paper, we assume a general graph description of a recommendation that a user's decisions are influenced by their inherent preferences, the recommender system, and their previous decisions. Without the intervention of recommendation system, users select which item to purchase according to their own natural preference, and this purchase/rating/clicking behavior is collected as the observational/historical data. We use P to denote the user's preference. Since the sequential nature of the data, we use subscript to denote the timestamp. Thus P_t reflects the user's preference at time step t. We use S to denote the exposure of items to users. Specifically, under an interventional setting, it represents the system's recommendation; under an observational setting without recommendations, it means that users are exposed to the item from a different source. We use D to denote user's decision towards inquired items and user's decision towards the recommended items in an observational and interventional setting respectively. While in an online setting, there might be an additional causal link from D_t to S_{t+1}, in this work, we adhere to an offline training schedule and do not consider the causal relationship depicted by the yellow arrow.

3 Demystifying User's Decision-Making Through Causal Formulation

In this section, we demonstrate how to estimate the impact of various factors on users' decisions. Specifically, we adhere to the causal graph outlined in Figure 1 and illustrate how users' decisions are influenced by the recommender system, previous decisions, and their inherent preferences.

3.1 Problem Setting

From a causal perspective, estimating how a sequence of system recommendation affects user's decision at $T = t$ is equivalent to estimating the causal effect of the decision under intervention from the system.

$$\mathbb{P}(D_t|\mathrm{do}(S_t, S_{t-1}, ..., S_1), P_t). \tag{1}$$

We intervene only in system recommendations to improve the purchasing likelihood under the recommendations. We cannot directly change a user's behavior or purchasing logic; instead, we aim to increase the probability of purchase through our recommendations.

From Figure 1, we observe that the decision D_t is directly influenced by the recommender system S_t at time t, the user's preference P_t, and the user's previous decision D_{t-1}. In a sequential recommendation setting, D_{t-1} inherently includes information about the recommender system at the prior time step S_{t-1}, the user's preference P_{t-1}, and the decision D_{t-2}. Therefore, while the most recent decision is a direct cause, the entire history can indirectly affect the user's current decision.

3.2 Estimating the Interventional Distribution.

To estimate such an interventional distribution, we need to (1) sequentially recover the causal relationship at the current timestamp and (2) maintain such causal structure in the training of a recommender system. We start with revealing the causal dependency between two subsequent decisions under a series system recommendations in the following theorem, which naturally reveals a constraint to maintain the causal structure.

Theorem 3.1. *At time $T = t$, the probability distribution for a user's decision given precedent recommendations $do(S_{t-1}, ..., S_1)$ and the preference P_t follows the equality*

$$\mathbb{P}(D_t|do(S_t, S_{t-1}, ..., S_1), P_t) = \sum_{D_{t-1}} \mathbb{P}(D_t|S_t, D_{t-1}, P_t) \cdot \mathbb{P}(D_{t-1}|do(S_{t-1}, ..., S_1), P_t) \tag{2}$$

The proof for the theorem is shown in Appendix C.

Theorem 3.1 describes how user's current decision is affected by previous decisions, the system and the user's preference. To explain this theorem in details, we write this summation for time $t = 2$, denoting the system recommended items as s_1, s_2 for $t = 1$ and $t = 2$ respectively. For simplicity, we consider $D_t \in \{0, 1\}$ where 0 denotes a negative decision and 1 denotes a positive decision. We can write

$$\begin{aligned}\mathbb{P}(D_2|\mathrm{do}(S_2 = s_2, S_1 = s_1), P_2) = &\sum_{D_1 \in \{0,1\}} \mathbb{P}(D_2|S_2 = s_2, D_1, , P_1)\mathbb{P}(D_1|\mathrm{do}(S_1 = s_1), P_1) \\ &= \mathbb{P}(D_2|D_1 = 1, S_2 = s_2, P_2)\mathbb{P}(D_1 = 1|\mathrm{do}(S_1 = s_1), P_1) + \\ &\quad \mathbb{P}(D_2|D_1 = 0, S_2 = s_2, P_2)\mathbb{P}(D_1 = 0|\mathrm{do}(S_1 = s_1), P_1).\end{aligned}$$

Here $do(\cdot)$ represents the interventional distribution where the recommender system controls the exposure as stated in Appendix B, and $\mathbb{P}(D_2|D_1 = 1, S_2 = s_2, P_2)$ represents the observational distribution such that the source of exposure is not the recommender system.

Hence, the estimation of this probability is equivalent to the prediction of one user's reaction to the recommended item s_2, taking their decision on s_1 into consideration, and weighted by the corresponding interventional probability. So long as this equality maintained at each timestamp, the causal relationship remains intact throughout the entire recommendation process.

3.3 Maximizing Users' Positive Decisions.

A general goal of a recommender system is to maximize the probability of users' positive decisions, e.g., purchase or click rate. Existing works equate such a goal as selecting the item that has the highest preference score for users. We on the other hand, consider to maximize the purchasing probability by adjusting system's behavior. This subtle distinction enables us to first generalize the concept and then incorporate Theorem 3.1 to maintain the inherent causal structure.

Instead of examining the user's preference towards each product, we compute the probability of positive decisions given a series of recommendations $\text{do}(S_t, S_{t-1}, ..., S_1)$, together with their current preference P_t. Hence, we can formulate the task at time t as an optimization problem:

$$s_t = \arg\max_{\text{do}(S_t)\in\mathcal{S}} \mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t). \tag{3}$$

Note that an alternative way to define the optimization problem is to consider the cumulative decisions $\sum_{i=1}^{T} D_i$. For simplicity we consider a one-step maximization as Equation (3) only, but the framework can be generalized to cumulative decisions easily.

To preserve the causal structure, we refer to Theorem 3.1 and accordingly propose CSRec (**C**ausal **S**equential **Rec**ommender) with this new optimization goal as the following:

$$\max_{\text{do}(S_t)\in\mathcal{S}} \mathbb{P}(D_t|\text{do}(S_t, ..., S_1), P_t) \tag{4}$$

$$\text{s.t.} \quad \mathbb{P}(D_t|\text{do}(S_t, ..., S_1), P_t) \;=\; \sum_{D_{t-1}} \mathbb{P}(D_t|S_t, D_{t-1}, P_t) \cdot \mathbb{P}(D_{t-1}|\text{do}(S_{t-1}, ..., S_1), P_t) \tag{5}$$

Remark 3.2. Concretely, CSRec considers optimizing the probability of users' positive decision rate by incorporating the causal relationship for a variety of factors.

3.4 Enforcing the constraint.

To enforce the constraint Equation (5), we need to estimate quantities from both the interventional distribution $\mathbb{P}(D_{t-1}|\text{do}(S_{t-1}, ..., S_1), P_t)$ and observational distribution $\mathbb{P}(D_t|S_t, D_{t-1}, P_t)$. Here we emphasize the difference between these two distributions. The $do(\cdot)$ notation in $\mathbb{P}(D_{t-1}|\text{do}(S_{t-1}, ..., S_1), P_t)$ refers to situations where the exposure is completely controlled by recommendation while S_t in observational distribution $\mathbb{P}(D_t|S_t, D_{t-1}, P_t)$, where S_t represents to items exposed to users from a different source than a recommender system.

Hence, probability $\mathbb{P}(D_t|S_t, D_{t-1}, P_t)$ can be understood as users' natural decision behavior D_t towards items S_t based on their previous decision D_{t-1} and temporal preference P_t. This is a natural goal for a traditional neural network based recommendation system such as SASRec [13] or BSARec [27]. Therefore, we estimate this quantity using an neural network recommender system trained on observational dataset.

Putting everything together, we build a recommender system that models users' decisions under the intervention of recommendations, parameterized as $f_t(\theta) := \mathbb{P}(D_t|\text{do}(S_t, ..., S_1), P_t)$. Consequently, Equation (5) can be written as

$$f_t = \sum_{D_{t-1}} \mathbb{P}(D_t|S_t, D_{t-1}, P_t) f_{t-1}. \tag{6}$$

We adapt a sequential recommender system trained on historical data, and parameterize it as $\tilde{f}$:

$$\mathbb{P}(\cdot|S_t, D_{t-1}, P_t) = \tilde{f}(D_{t-1}, ..., D; \eta). \tag{7}$$

With the general objective function of existing sequential recommender systems, we trained our model using the following objective

$$\mathcal{L}_t = -(D_t \log f_t(\theta) + (1 - D_t)\log(1 - f_t(\theta))), \text{s.t. } f_t(\theta) = \sum_{D_{t-1}} f_{t-1}(\theta)\tilde{f}(\eta). \tag{8}$$

By the Lagrangian multiplier method, we write the loss function for optimizing $f_t(\theta)$ as

$$\mathcal{L}_t = -(D_t \log f_t(\theta) + (1 - D_t)\log(1 - f_t(\theta))) + \left\| f_t(\theta) - \sum_{D_{t-1}} f_{t-1}(\theta)\tilde{f}(\eta) \right\|_2. \tag{9}$$

In summary, we first train a pre-selected sequential recommender system on the observational dataset and use that to estimate the quantity $\tilde{f} = \mathbb{P}(\cdot|S_t, D_{t-1}, P_t)$. Then we trained our model on interventional dataset to estimate $\mathbb{P}(D_{t-1}|\text{do}(S_{t-1}, ..., S_1), P_t)$ using the objective function Equation (9).

We note here that we can use any recommender system to estimate $\tilde{f} = \mathbb{P}(\cdot|S_t, D_{t-1}, P_t)$ so long as its goal is to estimate users' natural decision behavior. We show in Section 4 the performance of CSRec integrated with four different sequential recommender systems.

3.5 Estimate System's Influence Through Treatment Effect

With the decision-making process explicitly characterized as shown in Section 3, we can approximate the potential outcome for each candidate item to recommend, specifically the probability that the recommendation will be accepted, through *Treatment Effect Estimation*. For example, in the exploration and exploitation dilemma, people must predict the risk of recommending a novel item that is out-of-the-distribution. Additionally, when a system is trying to gather information about a new user, it is crucial to quickly understand the user without over-exploiting their trust in the system. Quantifying the system's influence on users can significantly enhance the development of a personalized and trustworthy system, moving beyond merely predicting user preferences from static data.

Treatment in causal inference refers to the action that applies (exposes, or subjects) to a unit[24]. For each unit-treatment pair, the outcome of the treatment when applied to the unit is referred as the potential outcome. With N treatments $T = \{1, 2, 3, ..., N\}$, the potential outcome of applying treatment T_i is denoted as $Y(T = T_i)$. The treatment effect can be quantitatively defined as follow.

Definition 3.3 (Binary Average Treatment Effect(ATE))**.** Suppose we want to measure the treatment effect of a treatment $T = 1$. Then the average treatment effect is defined as:

$$\mathbb{E}[Y(T = 1) - Y(T = 0)] \tag{10}$$

where $Y(T = 1)$ and $Y(T = 0)$ denote the potential treated and control outcome of the whole population respectively.

In the recommendation scenario, we extend Equation (21) to the difference between the user's decision with and without the recommendation.

Definition 3.4 (Treatment Effect in Recommendation)**.**

$$\text{TER}_t := \mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t) - \mathbb{P}(D_t|S_t, S_{t-1}, ..., S_1, P_t). \tag{11}$$

Here S_t represents the item exposed to users at timestamp t. In the observational setting ($\mathbb{P}(D_t|S_t, S_{t-1}, ..., S_1, P_t)$) the exposure is controlled by users or randomized, while in the interventional settings ($\mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t)$), it is controlled by the recommender system.

We compute the first term using our proposed model CSRec and the second term using any commonly used recommender systems:

$$\text{TER}_t(V) = f_t(V, X_{\text{intv}}; \theta) - \tilde{f}(V, X_{\text{obs}}; \eta) \tag{12}$$

where V denotes the set of all items, X_{intv} and X_{obs} represent interventional data and observational data respectively.

4 Experiment

4.1 Basic setup

4.1.1 Datasets and Baselines

Datasets. we evaluate CSRec on two datasets: a real-world Spotify Sequential Music dataset[1], and a synthetic book dataset, GPT4-books where the dataset details can be found in Appendix D and Appendix D.2.

Baselines. We compare CSRec with various sota baselines with different backbones: (1) RNN or CNN-based sequential models: GRU4Rec [11] and Caser [30] and (2) transformer-based sequential recommender systems: SASRec [13], BERT4Rec [29], FMLPRec [37], and BSARec [27]. We further show the performance of the proposed methods when integrated with existing recommendation models.

4.1.2 Evaluation metrics

We evaluate CSRec and the baselines models on both observational data where exposure to items does not come from recommendations and the interventional data where recommender system controls the exposure. On the observational data, we compare their performance in the task of forecasting the user's subsequent action from prior actions. Specifically, we compare CSRec and SR baselines via widely used Top-k metrics, HR@k (Hit Rate) and NDCG@k (Normalized Discounted Cumulative Gain) where k is set to 5, 10, and 20 [16]. While on the interventional data, we estimate the user's decision regarding a recommendation, and we compare the predicted purchase probability with the ground truth decisions via an adjusted hit rate defined as follows

$$\text{AHR}_{\text{CSRec}} = \frac{1}{N}\sum_{N} \delta(d_t, \hat{d}_t), \quad \hat{d}_t = \chi_{\geq 1-\alpha}(z_t), \tag{13}$$

where z_t is the model output, α is a pre-defined threshold; $\delta(a, b) = 1$ when $a = b$ and zero otherwise; $\chi_A(x)$ is an indicator function such that it equals to 1 when $x \in A$ and zero otherwise. For more details on the evaluation metric, see appendix D.2.1.

Table 1: Recommendation performance comparison of different models on GPT4-books dataset. X_{obs} denotes the observational data and evaluated using HR@k and NDCG@k for $k = 5, 10, 20$. X_{intv} denotes the interventional data and evaluated using AHR@α for $\alpha = 0.2, 0.5$. Boldface denotes the highest score among all models.

Type	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	FMLPRec	BSARec	CSRec
X_{obs}	↑ HR@5	0.0492	0.0489	**0.0533**	0.0528	0.0496	0.0524	0.0513
	↑ HR@10	0.1018	0.1001	**0.1057**	0.1055	0.1019	0.1035	0.1035
	↑ HR@20	0.1988	0.2014	0.2017	0.2011	0.2008	**0.2062**	0.2070
	↑ NDCG@5	0.0292	0.0285	0.0313	**0.0320**	0.0297	0.0316	0.0297
	↑ NDCG@10	0.0460	0.0448	0.0480	**0.0488**	0.0460	0.0479	0.0463
	↑ NDCG@20	0.0701	0.0701	0.0720	0.0726	0.0707	**0.0736**	0.0721
X_{intv}	↑ AHR@0.2	0.2431	0.2431	0.2438	0.2430	0.2435	0.2405	**0.932**
	↑ AHR@0.5	0.4967	0.4942	0.4981	0.5068	0.5006	0.5037	**0.932**

4.2 Experimental results

4.2.1 [RQ1]. The proposed method outperforms baselines on interventional data.

Table 1 and Table 2 show the performance of CSRec and baseline SR models on both interventional and observational data for datasets GPT4-books and MSSD. Here for CSRec, Table 1 shows the result of using FMLPRec for approximating the quantity $\mathbb{P}(D_t|S_t, D_{t-1}, P_t)$, and in Table 2, we used BSARec to estimate it. We used the adjusted hit rate (AHR) with two different ratios 0.2 and 0.5 as evaluation metrics. As a comparison, we also included the results for WSDM winning model (Few-Shot) and integrated it with our framework (CS-Few-Shot) on real-world MSSD, shown in Table 2.

Table 2: Recommendation performance comparison of different models on MSSD dataset. X_{obs} denotes the observational data and evaluated using HR@k and NDCG@k for $k = 5, 10, 20$. X_{intv} denotes the interventional data and evaluated using AHR@α for $\alpha = 0.2, 0.5$. Boldface denotes the highest score among all models. We compare sota sequential reccomenders and also the winning model, Few-Shot[2] of the WSDM challenge where MSSD is originally released. Few-Shot has a specialized design that fully considers the properties of music tracks.

		General SOTA Sequential Recommenders							**WSDM winning model**	
Data	Metric	GRU4Rec	Caser	SASRec	BERT4Rec	FMLPRec	BSARec	CSRec	Few-Shot	CS-Few-Shot
X_{obs}	↑ HR@5	0.1045	0.0963	0.0845	0.0219	0.1487	0.1736	**0.1745**	**0.1865**	0.1798
	↑ HR@10	0.1311	0.1012	0.1082	0.0407	0.1952	0.1901	**0.2007**	0.2235	**0.2364**
	↑ HR@20	0.1944	**0.3031**	0.2110	0.0769	0.2747	0.2878	0.2871	0.3712	**0.3731**
	↑ NDCG@5	0.0148	0.0286	0.0429	0.0341	0.0745	0.1039	**0.1041**	**0.1336**	0.1325
	↑ NDCG@10	0.0312	0.0457	0.0981	0.0771	0.1031	0.1487	**0.2117**	**0.1674**	0.1572
	↑ NDCG@20	0.0773	0.0717	0.1227	0.0985	0.1592	0.2793	**0.2801**	0.3412	**0.3435**
X_{intv}	↑ AHR@0.2	0.1117	0.1461	0.1926	0.1468	0.1412	0.1692	**0.3509**	0.4127	**0.6764**
	↑ AHR@0.5	0.2403	0.2467	0.2956	0.3401	0.3009	0.4323	**0.5371**	0.5061	**0.7418**

We observe that on both datasets, AHR for α at both 0.2 and 0.5 has been improved tremendously. On GPT4-books dataset, SASRec has the highest score for AHR@0.2, and CSRec approximately triples the score on GPT4-books, meaning that using $\alpha = 0.2$, the accuracy for predicting users' decisions using CSRec could be almost three times than that using baseline SR models. Even with a more relaxed metric AHR@0.5, on GPT4-books, baseline SR models is only doing slightly better than a random guess.

On real-world MSSD data, CSRec outperforms SOTA SR models by a factor of 1.5 using AHR@0.2, with a higher score on AHR@0.5 as well. Furthermore, when integrated with WSDM winning model, our framework (CS-Few-Shot) achieves the hit rate accuracy as high as 0.6764 using conversion threshold $\alpha = 0.2$. Compared to the original model, our framework is able to improve the accuracy of modeling users' decision by 64%.

It is because baseline SR models do not take negative decisions explicitly into consideration. Therefore, they neglected the fact that current decision is affected by previous recommendations (either successful or unsuccessful). This observation highlights the significance of distinguishing users' natural selection processes and their decision making processes under recommendation.

4.2.2 [RQ2]. The proposed method does not degrade the performance on observational data.

We compared CSRec and baseline SR models on the observational data for both GPT4-books and MSSD. Results are shown in Table 1 and Table 2 respectively.

On MSSD dataset (Table 2), we observe that compared to the general SOTA SR models, CSRec achieves higher score for the five metrics out of six except for HR@20. Moreover, when integrating with the winning model, our framework (CS-Few-Shot) is able to maintain its existing level of performance. On GPT4-books dataset (Table 1), though CSRec does not improve the baseline SR models on observational data, it also does not cause any degradation on the performance. Specifically, on GPT4-books, CSRec achieved a slightly higher score on HR@20 than BSARec, which has the highest score among all baselines. In general, CSRec is able to receive median scores for all metrics.

Consequently, CSRec is able to improve existing models on interventional data while maintaining their performance on observational data. This reveals that CSRec has a wide application in practice.

5 Conclusion

In this paper, by distinguishing between user natural selection process and their decisions making under recommendation, we propose a framework to model the causal relationship between user's decision and various factors including their natural preference and their previous decision towards recommendations. We introduce CSRec, a general framework with straightforward implementation that can easily adapted to SOTA sequential recommenders. Experimental results show that our framework is capable of improving the accuracy of predicting users' decision under recommendations without causing any degradation on non-interventional setting.

References

[1] Brian Brost, Rishabh Mehrotra, and Tristan Jehan. The music streaming sessions dataset. In The World Wide Web Conference, pages 2594–2600, 2019.

[2] Sungkyun Chang, Seungjin Lee, and Kyogu Lee. Sequential skip prediction with few-shot in streamed music contents. arXiv preprint arXiv:1901.08203, 2019.

[3] Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. Autodebias: Learning to debias for recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 21–30, 2021.

[4] Jiuhai Chen and Jonas Mueller. Automated data curation for robust language model fine-tuning. arXiv preprint arXiv:2403.12776, 2024.

[5] Jiuhai Chen, Rifaa Qadri, Yuxin Wen, Neel Jain, John Kirchenbauer, Tianyi Zhou, and Tom Goldstein. Genqa: Generating millions of instructions from a handful of prompts. arXiv preprint arXiv:2406.10323, 2024.

[6] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In Proceedings of the eleventh ACM conference on recommender systems, pages 152–160, 2017.

[7] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guanfeng Liu, Yanchi Liu, and Victor S Sheng. Frequency enhanced hybrid attention network for sequential recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 78–88, 2023.

[8] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. Causal inference in recommender systems: A survey and future directions. ACM Transactions on Information Systems, 42(4):1–32, 2024.

[9] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. Understanding echo chambers in e-commerce recommender systems. In Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pages 2261–2270, 2020.

[10] Ruining He, Wang-Cheng Kang, and Julian McAuley. Translation-based recommendation. In Proceedings of the eleventh ACM conference on recommender systems, pages 161–169, 2017.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2016.

[12] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. Preference amplification in recommender systems. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 805–815, 2021.

[13] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In 2018 IEEE International Conference on Data Mining (ICDM), pages 197–206, 2018. doi: 10.1109/ICDM.2018.00035.

[14] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In 2018 IEEE international conference on data mining (ICDM), pages 197–206. IEEE, 2018.

[15] Anastasiia Klimashevskaia, Dietmar Jannach, Mehdi Elahi, and Christoph Trattner. A survey on popularity bias in recommender systems. User Modeling and User-Adapted Interaction, pages 1–58, 2024.

[16] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. Commun. ACM, 65(7):75–83, jun 2022. ISSN 0001-0782. doi: 10.1145/3535335. URL https://doi.org/10.1145/3535335.

[17] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. Text is all you need: Learning language representations for sequential recommendation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 1258–1267, 2023.

[18] Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. Recent developments in recommender systems: A survey. IEEE Computational Intelligence Magazine, 19(2):78–95, 2024.

[19] Ruibo Liu, Jerry Wei, Fangyu Liu, Chenglei Si, Yanzhe Zhang, Jinmeng Rao, Steven Zheng, Daiyi Peng, Diyi Yang, Denny Zhou, and Andrew M. Dai. Best practices and lessons learned on synthetic data, 2024. URL https://arxiv.org/abs/2404.07503.

[20] Xiaoyu Liu, Hanlin Lu, Jianbo Yuan, and Xinyu Li. Cat: Causal audio transformer for audio classification. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[21] Xiaoyu Liu, Paiheng Xu, Junda Wu, Jiaxin Yuan, Yifan Yang, Yuhang Zhou, Fuxiao Liu, Tianrui Guan, Haoliang Wang, Tong Yu, et al. Large language models and causal inference in collaboration: A comprehensive survey. arXiv preprint arXiv:2403.09606, 2024.

[22] Xiaoyu Liu, Jiaxin Yuan, Bang An, Yuancheng Xu, Yifan Yang, and Furong Huang. C-disentanglement: discovering causally-independent generative factors under an inductive bias of confounder. Advances in Neural Information Processing Systems, 36, 2024.

[23] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pages 188–197, 2019.

[24] Judea Pearl. The do-calculus revisited. arXiv preprint arXiv:1210.4852, 2012.

[25] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. Contrastive learning for representation degeneration problem in sequential recommendation. In Proceedings of the fifteenth ACM international conference on web search and data mining, pages 813–823, 2022.

[26] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In international conference on machine learning, pages 1670–1679. PMLR, 2016.

[27] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An attentive inductive bias for sequential recommendation beyond the self-attention, 2024.

[28] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. An attentive inductive bias for sequential recommendation beyond the self-attention. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, pages 8984–8992, 2024.

[29] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer, 2019.

[30] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding, 2018.

[31] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. Sequential recommender systems: challenges, progress and prospects. arXiv preprint arXiv:2001.04830, 2019.

[32] Yixin Wang and Michael I Jordan. Desiderata for representation learning: A causal perspective. arXiv preprint arXiv:2109.03795, 2021.

[33] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommender systems. In Fourteenth ACM Conference on Recommender Systems, pages 426–431, 2020.

[34] Shuyuan Xu, Yingqiang Ge, Yunqi Li, Zuohui Fu, Xu Chen, and Yongfeng Zhang. Causal collaborative filtering. arXiv preprint arXiv:2102.01868, 2021.

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

[36] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 367–377, 2021.

[37] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In Proceedings of the ACM Web Conference 2022, WWW '22. ACM, April 2022. doi: 10.1145/3485447.3512111. URL http://dx.doi.org/10.1145/3485447.3512111.

[38] Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. Attention calibration for transformer-based sequential recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, pages 3595–3605, 2023.

[39] Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. Calibrated self-rewarding vision language models. arXiv preprint arXiv:2405.14622, 2024.

[40] Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. Explore spurious correlations at the concept level in language models for text classification, 2024. URL https://arxiv.org/abs/2311.08648.

[41] Yuhang Zhou, Jing Zhu, Paiheng Xu, Xiaoyu Liu, Xiyao Wang, Danai Koutra, Wei Ai, and Furong Huang. Multi-stage balanced distillation: Addressing long-tail challenges in sequence-level knowledge distillation. arXiv preprint arXiv:2406.13114, 2024.

A Do-Calculus

Rule A.1 (Insertion/deletion of observations)**.**

$$P(y|do(x), z, w) = P(y|do(x), w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\overline{X}}} \tag{14}$$

Rule A.2 (Action/observation exchange)**.**

$$P(y|do(x), do(z), w) = P(y|do(x), z, w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\overline{X}\underline{Z}}} \tag{15}$$

Rule A.3 (Insertion/deletion of actions)**.**

$$P(y|do(x), do(z), w) = P(y|do(x), w) \quad \text{if} \quad (Y \perp\!\!\!\perp Z|X, W)_{G_{\overline{X}\,\overline{Z(W)}}} \tag{16}$$

where $G_{\overline{X}}$ is the graph with all incoming edges to X being removed, $G_{\underline{W}}$ is the graph with all outcoming edges to W being removed, and $Z(W)$ is the set of Z-nodes that are not ancestors of any W-node.

Intuitively, Rule A.1 states when an observant can be omitted in estimating the interventional distribution, Rule A.2 illustrates under what condition, the interventional distribution can be estimated using the observational dataset, and Rule A.3 decides when we can ignore an intervention.

B Background and Preliminary

The goal of training a recommender system varies across different contexts, such as maximizing the likelihood of a recommended item being accepted or enhancing long-term trust and user satisfaction. At the heart of developing any recommender system is the need to understand how users make decisions and how these decisions can be influenced.

While existing research has made significant strides in recovering users' natural preferences from observational data, two critical aspects are under-emphasized in sequential recommendation: (1) **System-Controlled Exposure**: Unlike users' spontaneous purchases, recommender systems control how items are exposed to users. It is not uncommon for users' purchase decisions to be adversely affected by system recommendations. However, this influence is often overlooked by existing formulations due to differences in the underlying training goals. (2) **Collaborative Influence on Decision-Making**: Users' decisions are shaped by various recommendation scenarios and design needs, necessitating a unified and systematic approach to analyze, understand, and evaluate these influences.

In this section, we begin by reviewing how existing methods approach recommendation. We then reformulate the recommendation process from a causal perspective, introducing the basic concepts of causal inference and demonstrating how to evaluate causal relationships using a causal graph. Finally, we apply these methods to analyze and model the causal relationships within subsequent purchases, as discussed in section 2 and section 3.

Recommendation as Sequential Prediction: The Traditional Viewpoint. Sequential recommender systems model historical user interactions as temporally-ordered sequences to recommend potential items that users are interested in. In the general formulation as a sequential dataset, we are given an item set $\mathcal{I}$ and a user's interaction sequence $s = \{i_1, i_2, ..., i_n\}$, $i \in \mathcal{I}$ in temporal order where n is the length of the sequence[17]. This formulation only accounts for cases where all items in the sequence are accepted by users. However, during the recommendation process, users may also be recommended items in which they are not interested. In other words, if we distinguish between exposure and acceptance, with acceptance denoted as 1 and rejection as 0, the traditional viewpoint only models sequences composed entirely of 1s, thereby overlooking sequences that include 0s.

Recommendation from a causal perspective. Compared to the existing formulation that focuses on uncovering users' unbiased interests, causal inference methods offer a more general and intuitive approach by aiming to characterize how users' decisions are causally influenced. In this context, the recommender system acts as an *intervention* on the user, specifically by controlling the exposure of items to the user, which naturally separates the concepts of exposure and acceptance.

Causal graph through DAG. The causal relationships among variables can be represented by a Directed Acyclic Graph (DAG). In this graph, each (potentially high-dimensional) variable is represented as a node, and the directed edges indicate causal relationships, always pointing from parent nodes to child nodes. A causal graph can be estimated using do-calculus as detailed in Appendix A.

Intervention and do-operator. *Intervention* is one of the fundamental concepts in causal inference. When we intervene on a variable, we set the value of a variable and cut all incoming causal arrows since its value is thereby determined only by the intervention [24]. The intervention is mathematically represented by the *do-operator* $do(\cdot)$. We include the do-calculus for identifying causal effects from a causal graph in appendix A. In the context of recommendation, we are trying to understand how the *changes in recommended items affect the user's decision and satisfaction*, which can be represented by

Definition B.1 (Recommendation in Causal Inference)**.** Let D and S represents the user decision and the system recommendation respectively,

$$P(D|do(S=s),P) \tag{17}$$

characterizes an interventional distribution and reflects how a change of S affects the distribution of D for users represented by the preference P .

Observational distribution and interventional distribution. Observational distributions refer to data passively collected without any explicit manipulation, reflecting correlations influenced by potential underlying factors. In contrast, interventional distributions are generated through controlled interventions, where specific variables are deliberately manipulated to observe causal outcomes. In this paper, since we are interested in exploring how system intervention affect user's purchase decision, we refer to historical data, where user selects what to buy, as the observational data, and refer to the data which contains user's reaction towards recommender system as the interventional data.

Treatment effect estimation. Treatment effect estimation is a fundamental aspect of causal inference that involves quantifying the impact of an intervention on an outcome. It measures the difference in outcomes resulting from a specific treatment compared to what would have occurred in the absence of the treatment. This method offers a rigorous approach to evaluating the influence of recommender systems on user decisions. By assessing the variance in user decisions before and after receiving recommendations, we can accurately quantify the system's impact on user behavior.

C Proof for Theorem 3.1

Proof. Suppose that the user's decision D_t follows the causal relationship in Figure 1, then we can write

$$\mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t) = \sum_{D_{t-1}} \mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t, D_{t-1}) \cdot \mathbb{P}(D_{t-1}|\text{do}(S_t, S_{t-1}, ..., S_1), P_t) \tag{18}$$

From the figure we see that there's no causal influence of S_t on the decision D_{t-1}, hence we can write

$$\mathbb{P}(D_{t-1}|\text{do}(S_{t-1}, ..., S_1), P_t)$$

We know that D_t and S_t satisfies

$$(D_t \perp\!\!\!\perp S_t|S_{t-1}, ..., S_1, P_t, D_{t-1})_{G_{\overline{S_{t-1},...,S_1},\underline{P_t,D_{t-1}}}}.$$

Hence, using Rule A.2, we can obtain

$$\mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t, D_{t-1}) = \mathbb{P}(D_t|S_t, \text{do}(S_{t-1}, ..., S_1), P_t, D_{t-1}).$$

From the causal graph, we can also see that

$$(D_t \perp\!\!\!\perp \{S_{t-1}, ..., S_1\}|S_2, D_{t-1})G_{\overline{\{S_{t-1},...,S_1\}(S_t,D_{t-1})}}.$$

Therefore, Rule A.3 leads to

$$\mathbb{P}(D_t|S_t, \text{do}(S_{t-1}, ..., S_1), P_t, D_{t-1}) = \mathbb{P}(D_t|S_t, P_t, D_{t-1})$$

Combining the results for the two component we proved the result for Theorem 3.1. □

D Experimental Details

D.1 Spotify: Sequential Music Streaming Challenge

MSSD is first published at the WSDM Cup 2019 Spotify Sequential Skip Prediction Challenge [1]. It has approximately 160 million listening sessions, associated user actions, and 3.7 million tracks with acoustic features. The task is to predict whether a particular user will skip individual tracks encountered in a listening session. The sessions in this dataset are sampled from various sources such as radio, personalized recommendations, users' own collections, expertly curated playlists, and contextual but non-personalized recommendations. Records of listening sessions from users' own collection approximate the observational dataset while the personalized recommendation source serves as an ideal interventional dataset for studying the user-system dynamics from a causal perspective: users who listen to personalized recommendations do not have direct control over what they will hear. In other words, the exposure is exogenously determined by the RecSys, which reflects Spotify's belief in the users' preferences. In the remainder of the paper, we focus on the listening sessions collected from such a personalized recommendation setting as the interventional data, and the user's own collection as the observational data.

Following the causal perspective of recommendation, the treatment is considered effective if the user does not skip it and the treatment is presented as a ranked list of the songs selected for the users. Therefore, we define two tasks of interest on MSSD: skip prediction and track ranking. Skip prediction measures the individual treatment effect of each track for a certain user, while we measure the relative effectiveness among all treatments when ranking the tracks. Skip prediction is equivalent to estimating treatment effect, and the track ranking corresponds to selecting the most effective treatments. To preserve user privacy, Spotify randomly cut the listening sessions in MSSD from longer sessions so that a single session contains no more than 20 songs.

Table 3: Results for CSRec integrated with baselines SASRec, BERT4Rec, FMLPRec and BSARec on GPT4-books using AHR@0.2 and AHR@0.5. CSRec + <model> represents the performance of CSRec integrated with baseline <model>.

Metric	SASRec	CSRec + SASRec	BERT4Rec	CSRec + BERT4Rec	FMLPRec	CSRec + FMLPRec	BSARec	CSRec + BSARec
↑ AHR@20	0.2438	0.3914	0.2430	0.3046	0.2435	0.932	0.2405	0.925
↑ AHR@50	0.4981	0.9176	0.5068	0.9153	0.5006	0.932	0.5037	0.932

D.2 GPT4-books

Given the limited access to interventional data, where the recommender system fully controls the exposure. Previous work has explored the potential of using a large language model agent to generate synthetic data for downstream tasks [19; 40; 41; 5; 4; 39]. To resolve the data shortage, we also created a synthetic dataset using GPT-4 as agents to mimic users' reactions to certain recommendations. Specifically, we use GPT4-turbo as the agent that mimics users with various tastes of books.

Preparing the item list. We first sampled a list of book names from the Amazon Book dataset[23]. We use books as items to recommend because GPT4 has had prior information towards these items. We then asked the GPT4 about the basic information about the books. The reason is that we want to validate the GPT4 knows the books and is be able to serve as potential readers. Specifically we validate that by asking the GPT about basic information and keywords about the book names that we provided and comparing them with what we have in the Amazon Book dataset.

Synthetic users. The second step is to create synthetic users. We refer to the scale of similar datasets used by existing baselines[11; 30; 27; 29; 37] and define 10000 users. Each user is defined by a probability distribution of preference across different genres. For example, we describe a user as who's preference towards history, fiction, novels, and romance is $25\%, 25\%, 25\%, 25\%$ where the probabilities are randomly generated.

Synthetic observational/interventional dataset. For each user, we asked GPT4 to generate the synthetic purchase records of various length. Specifically, we collect the top 100 popular books for each genre, and randomly set a length l for each user. We then asked the GPT4 to generate a sequence of length l,which is the most possible purchase records for this user, among the list of books. For the interventional dataset, instead of asking GPT4 what is the most possible purchase records, we generate random sequences of recommendations, and asked the agent whether the recommended book will be purchased, and the probability of such a decision. Timestamp information is also provided to GPT4.

D.2.1 Evaluation metrics

We evaluate CSRec and the baselines models on both observational data where exposure to items does not come from recommendations and the interventional data where recommender system controls the exposure. On the observational data, we compare their performance in the task of forecasting the user's subsequent action from prior actions. While on the interventional data, we estimate the user's decision regarding a recommendation. Given the difference in the objective that our model and other recommender system aim to achieve, necessary adaptation is required and we provide the detailed procedure as follows.

Observational data. On observational data, we compare CSRec and SR baselines via widely used Top-k metrics, HR@k (Hit Rate) and NDCG@k (Normalized Discounted Cumulative Gain) where k is set to 5, 10, and 20 [16]. HR@k measures whether the ground-truth item is in the top-k positions of the recommendation list. NDCG on the other hand, takes the rank into considerations that it assigns higher scores to higher positions in the recommendation list.

We would like to point out a natural adjustment we make on CSRec for evaluation on observational data here. Even though CSRec is configured to model $\mathbb{P}(D_t|\text{do}(S_t, ..., S_1), P_t)$ under the interventional setting, on historical data, we can safely assume that all previous purchased items were recommended and accepted. The output probabilities are then ranked and evaluated using HR@k and NDCG@k.

Interventional data: On interventional data, the dataset contains two sequences: recommended items and users' corresponding decisions $\in \{0, 1\}$ indexed with time. We aim to examine how well models estimate user's decision when being recommended on interventional data.

For CSRec, the model outputs the predicted users' purchase probability for the next recommended items. We then compare this resulting probability with the ground truth decisions via an adjusted hit rate defined as follows

$$\text{AHR}_{\text{CSRec}} = \frac{1}{N}\sum_{N} \delta(d_t, \hat{d}_t), \quad \hat{d}_t = \chi_{\geq 1-\alpha}(z_t), \tag{19}$$

where z_t is the model output, α is a pre-defined threshold; $\delta(a, b) = 1$ when $a = b$ and zero otherwise; $\chi_A(x)$ is an indicator function such that it equals to 1 when $x \in A$ and zero otherwise.

Concretely, for a recommended item and ground truth decision $d_t \in \{0, 1\}$, CSRec outputs a predicted probability $z_t \in [0, 1]$. Then for a pre-defined α, we considered the predicted decision $\hat{d}_t$ as 1 if $z_t \geq 1 - \alpha$ and 0 otherwise. Then the accuracy is computed by counting the ratio of the succeeded prediction after this conversion.

For baseline models, we convert the model output into users' decisions based on the ranking of the recommended items. Notice that the baseline SR models do not take previous decisions as explicit input. Hence we filtered the interventional data sequence such that only recommended items with positive decisions are preserved. This strategy is similar to the post-processing procedure for FMLPRec [37] and BSARec [27].

To convert the ranking of the recommended items to users' purchasing decisions, we consider the users' decision to be 1 whenever the predicted score for recommended items are in the top-α-percent $\mathcal{A}_\alpha$. We then compute the binary cross entropy and adjusted hit rate by comparing the converted decisions with the ground truth decision. Mathematically, AHR for baselins are defined as follows:

$$\text{AHR}_{\text{baselines}} = \frac{1}{N}\sum_{N} \delta(d_t, \hat{d}_t), \quad \hat{d}_t = \chi_{\mathcal{A}_\alpha}(\tilde{z}_t). \tag{20}$$

Remark D.1. We computed AHR@α for both CSRec and baseline SR models through the discrepancy of their predicted decision after conversion and the ground truth decision. The difference lies in the conversion due to the difference of the objective of CSRec and baseline SR models. Specifically, results for CSRec are mapped to $\{0, 1\}$ according to the absolute value of the output because we model the probability; results for baseline SR models are mapped to $\{0, 1\}$ depending on the ranking for the inquired item.

D.3 Generalization of the causal graph used in this paper.

In this paper, we propose a novel approach to model the recommendation process from a causal perspective. We begin by constructing a causal graph that represents the recommendation scenario of interest. This graph allows us to identify how a user's decisions are influenced by various factors within the system. Rather than considering the sequential recommendation as a next-item prediction task, our approach focuses on understanding how a user's decisions can be positively influenced by specific factors.

As illustrated in fig. 1, the recommender system is conceptualized as an intervention that impacts the user's decision-making process. This causal graph is versatile and applicable to both intervention-based and observational scenarios. In the former, the recommender system acts as the direct intervention, while in the latter, the user's decisions are assumed to be influenced by unobserved sources. Thus, our model provides a framework for understanding the effects of both observed and unobserved interventions on user behavior.

D.4 Additional Results.

Table 4: Improvement of CSRec integrated with baselines SASRec, BERT4Rec, FMLPRec and BSARec on GPT4-books using AHR@0.2 and AHR@0.5. Each column δ_{model} represents the improvement in scores after applying CSRec on each model with the percentage change in the parenthesis.

Metric	δ_{SASRec}	δ_{BERT4Rec}	δ_{FMLPRec}	δ_{BSARec}
↑ AHR@0.2	0.1476 (60.5%)	0.0616 (25.5%)	0.6885 (283%)	0.6845 (285%)
↑ AHR@0.5	0.4195 (84.0%)	0.4085 (80.6%)	0.4314 (86.2%)	0.4283 (85.2%)

See Table 4

E Estimate System's Influence Through Treatment Effect

With the decision-making process explicitly characterized as shown in Section 3, we can approximate the potential outcome for each candidate item to recommend, specifically the probability that the recommendation will be accepted.

Beyond approximating the user's decision process, it is also essential to isolate and quantify the system's influence on its users. For example, in the exploration and exploitation dilemma, people must predict the risk of recommending a novel item that is out-of-the-distribution. Additionally, when a system is trying to gather information about a new user, it is crucial to quickly understand the user without over-exploiting their trust in the system. Quantifying the system's influence on users can significantly enhance the development of a personalized and trustworthy system, moving beyond merely predicting user preferences from static data. Estimating how one variable affects another is addressed by measuring the **treatment effect** in causal inference.

E.1 Treatment effect in recommendation

Treatment in causal inference refers to the action that applies (exposes, or subjects) to a unit[24]. For each unit-treatment pair, the outcome of the treatment when applied to the unit is referred as the potential outcome. With N treatments $T = \{1, 2, 3, ..., N\}$, the potential outcome of applying treatment T_i is denoted as $Y(T = T_i)$. The treatment effect can be quantitatively defined as follow.

Definition E.1 (Binary Average Treatment Effect(ATE))**.** Suppose we want to measure the treatment effect of a treatment $T = 1$. Then the average treatment effect is defined as:

$$\mathbb{E}[Y(T = 1) - Y(T = 0)] \tag{21}$$

where $Y(T = 1)$ and $Y(T = 0)$ denote the potential treated and control outcome of the whole population respectively.

In the recommendation scenario, we extend Equation (21) to the difference between the user's decision with and without the recommendation.

Definition E.2 (Treatment Effect in Recommendation)**.**

$$\text{TER}_t := \mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t) - \mathbb{P}(D_t|S_t, S_{t-1}, ..., S_1, P_t). \tag{22}$$

Here S_t represents the item exposed to users at timestamp t. In the observational setting ($\mathbb{P}(D_t|S_t, S_{t-1}, ..., S_1, P_t)$) the exposure is controlled by users or randomized, while in the interventional settings ($\mathbb{P}(D_t|\text{do}(S_t, S_{t-1}, ..., S_1), P_t)$), it is controlled by the recommender system.

We compute the first term using our proposed model CSRec and the second term using any commonly used recommender systems:

$$\text{TER}_t(V) = f_t(V, X_{\text{intv}}; \theta) - \tilde{f}(V, X_{\text{obs}}; \eta) \tag{23}$$

where V denotes the set of all items, X_{intv} and X_{obs} represent interventional data and observational data respectively.

We then demonstrate two cases where the quantified system's influence on users effectively broadens the perspective of building a recommendation system.

Evaluating Recommender System Quantifying the system's influence in a causal perspective could provide more insight into the effectiveness of the recommender system. It allows for understanding how recommendations change and shape users' decisions over time, beyond the single accuracy metrics. Moreover, by inspecting the treatment effect over different users (heterogeneous treatment effect), we can probe the mechanism of the recommender system and increase the interpretability.

Exploration and Exploitation The quantified system's influence on each timestamp could also contribute to building a balanced recommender system between exploration and exploitation. The system could more precisely identify which time to give the suggestions based on the existing knowledge (exploitation) and when to test the new items (exploration). This dynamic strategy accommodates the overall user experience and allows the discovery of new content.

F Related Work

Sequential Recommender Systems Sequential recommender systems (SRSs) learn the user-item interactions (e.g., view or purchase items on an online shopping platform) by modelling the dependencies in a sequence. Contrary to the traditional recommender systems which focus more on static learning through content-based and collaborative filtering[31], SRSs treat the interaction in a more dynamic way and hope to capture the inherent logic between the current and recent preference of a user. Traditional works formulate the task of sequential prediction as a first-order or high-order Markov Chain[10], whereas more recent works make use of the rapid development of the deep neural networks and propose RNN-based[6] and, more recently, attention-based[14; 28] frameworks to capture the temporal features. [25; 7] applies contrastive learning strategies to help the learning of the sequential dependencies, [38] adjusts unreliable attention weights for a more robust generalization. Despite the great success, there are two fundamental challenges that remain unresolved. However, the above works equate preference to decisions. However, user preference towards the item is only a partial reason for a purchase decision. For example, a user is highly unlikely to purchase another refrigerator no matter how they like it. Such interactive relationship is not fully captured or even modeled in existing works. In our work, we propose a framework that accurately characterizes the causal relationship between the recommendation from the system, the user's past purchasing behaviour and the user's preference, obtaining universally better performance.

Causal Inference in Recommendation. Causal inference methods are widely used in estimating the causal effect[24], understanding the compound causal relationship among various factors[20; 32], injecting external knowledge to the model[22], and isolating the causal relationship of certain variables. Causal inference is to estimate potential outcomes when treatment is applied[35; 21]. It originated from the scenario of prescribing medicine to patients. The recommendation is naturally a causal inference task when treating each item as a potential treatment. The goal is to predict the user's preference towards these treatments(i.e., the recommended items). There is a line of works framing recommendation as a causal inference task [8]. [26] show how estimating the quality of a recommendation system can be approached with Inverse Propensity-reWeighting(IPW) techniques commonly used in causal inference, and the proposed method is based on matrix factorization.[33] is also based on traditional matrix factorization. They make separate estimations to correct unobserved confounders (the factor that affects both treatment assignment and potential outcome). [34] propose a general framework for modelling causality in collaborative filtering and recommendation. [36] applies contrastive learning to solve the counterfactual reasoning problem. [3] derives a universal strategy for fixing various biases. Compared to traditional non-causal counterparts, causal recommender systems obtained successes in user-item interaction prediction and de-bias observational training data. In this paper, we propose the first causal framework for quantifying sequential recommendation and introduce a constraint that maintains the casual relationship in subsequent purchases in the modelling.