Published: 12 September 2025
Online AM: 15 April 2025
Accepted: 20 February 2025
Revised: 20 December 2024
Received: 15 September 2024

RESEARCH-ARTICLE

Causal Time-aware News Recommendations with Large Language Models

QIAN LI, Curtin University, Perth, WA, Australia

HAORAN YANG, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong

DIANER YU, University of Technology Sydney, Sydney, NSW, Australia

QING LI, The Hong Kong Polytechnic University, Hong Kong, Hong Kong, Hong Kong

GUANGDONG XU, The Education University of Hong Kong, Hong Kong, Hong Kong

Causal Time-aware News Recommendations with Large Language Models

SIRUI HUANG, University of Technology Sydney, Sydney, Australia and Hong Kong Polytechnic University, Hong Kong SAR, China
QIAN LI, School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, Australia
HAORAN YANG and DIANER YU, University of Technology Sydney, Sydney, Australia
QING LI, The Hong Kong Polytechnic University, Hong Kong SAR, China
GUANDONG XU, University of Technology Sydney, Sydney, Australia and The Education University of Hong Kong, Hong Kong SAR, Australia

Predicting user satisfaction over time is crucial in news recommendations, as users' preferences are significantly influenced by various time-variant factors. Traditional correlation-based recommenders often suffer from redundant relationships, which can undermine their effectiveness over time. This work takes a time-aware causal approach to news recommendations, treating exposed news at a predicted time as the treatment variable and the resulting user satisfaction as the outcome variable. Capturing the evolving causal effects of exposed news items on user satisfaction poses significant challenges, particularly stemming from the need to model complex dependencies among time-variant covariates, such as news popularity and recency, as well as to effectively leverage the inherent user preferences embedded in time-invariant covariates. To these ends, we propose the *CA*u*S*al *T*ime-aware *Rec*ommender, named *CAST-Rec*, which accounts for the causal influences of both time-variant and time-invariant covariates. Specifically, we model the intricate causal dependencies among time-variant covariates through a series of transformer-based causal blocks. For time-invariant covariates, we utilize the semantic understanding and generative capabilities of Large Language Models (LLMs) to infer inherent user preferences while mitigating potential confounding effects. Extensive experiments demonstrate the superior performance of CAST-Rec compared to various news recommendation models and across multiple LLM implementations.

CCS Concepts: • **Information systems** → **Personalization**; *Data mining*;

Additional Key Words and Phrases: Causal inference, news recommendations, sequential recommendation, large language models

This work was supported in part by the National Natural Science Foundation of China under Grants No. 62072257, the Australian Research Council Under Grants DP22010371 and LE220100078, and by the Hong Kong Research Grants Council under General Research Fund (Project No. 15200021), as well as Research Impact Fund (Project No. R1015-23).
Authors' Contact Information: Sirui Huang, University of Technology Sydney, Sydney, Australia and Hong Kong Polytechnic University, Hong Kong SAR, China; e-mail: sirui.huang@student.uts.edu.au; Qian Li (corresponding author), School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Perth, Australia; e-mail: qli@curtin.edu.au; Haoran Yang, University of Technology Sydney, Sydney, Australia; e-mail: haoran.yang-2@student.uts.edu.au; Dianer Yu, University of Technology Sydney, Sydney, Australia; e-mail: Dianer.Yu-1@student.uts.edu.au; Qing Li, The Hong Kong Polytechnic University, Hong Kong SAR, China; e-mail: qing-prof.li@polyu.edu.hk; Guandong Xu (corresponding author), University of Technology Sydney, Sydney, Australia and The Education University of Hong Kong, Hong Kong SAR, China; e-mail: Guandong.Xu@uts.edu.au.

ACM 1558-2868/2025/9-ART147
https://doi.org/10.1145/3729422

ACM Reference format:
Sirui Huang, Qian Li, Haoran Yang, Dianer Yu, Qing Li, and Guandong Xu. 2025. Causal Time-aware News Recommendations with Large Language Models. *ACM Trans. Inf. Syst.* 43, 6, Article 147 (September 2025), 25 pages.
https://doi.org/10.1145/3729422

1 Introduction

In an era of information overload, **News Recommendation Systems (NRSs)** play a vital role in delivering personalized content, enabling users to discover timely and relevant news amid the overwhelming volume of information. To deliver timely and personalized recommendations, NRSs are required to accurately model time-variant factors that capture evolving trends and user interests, while also accounting for time-invariant factors, such as the semantics of exposed news. This dual focus ensures that users receive updates on trending topics that align with their preferences, thereby enhancing the effectiveness of recommendations.

Time is indispensable in news recommendations, as it is crucial for ensuring that users receive the most current and relevant information. Multiple temporal factors, whether observed or not, can influence users' satisfaction with recommended items [25]. Attempting to account for all of these factors is impractical and would add considerable complexity [11]. Given the constraints on data availability, such as the lack of contextual user information due to privacy concerns, the recency and popularity of news items are widely used as two key time-variant factors in recommendations [22, 35]. As illustrated in Figure 1(a), news users are influenced by the recency and/or popularity of news items. The bold black arrow on the far left of the timeline indicates the release time of news items A and B. As time progresses from left to right, the recency of news A and B diminishes. The popularity of items diminishes as the color transitions from darker to lighter shades. In the example shown in Figure 1, the first user prefers fresh news, while the other prioritizes the popularity of news. Existing NRSs can be categorized by the time-variant factors they consider: recency [6, 28], popularity [1, 22], and both [8, 9, 31]. The first category of models addresses time-decaying correlations between exposed news and their recency from the user's perspective by distinguishing between short-term and long-term user preferences, leveraging fixed or adaptive time windows to prioritize recently interacted news while down-weighting outdated content [28]. Popularity-based models identify correlations between global popularity and exposed news by analyzing topic prevalence and prioritizing news within trending topics [1]. Methods that account for both recency and popularity consider the correlations of exposed news by incorporating these two factors [9, 53]. For example, in the user–item interaction graph in [9], recency of an item node can be encoded by node features, while the degree of the item node shows its popularity.

While these methods effectively approach time, they primarily rely on correlations and are, therefore, skewed by redundant associations. A more recent study [4] examines the causal impact of time on exposed news but simplifies time by treating it as a static variable. This approach does not capture the complex dependencies between multiple time-variant factors and overlooks how their causal influences on the user satisfactions evolve over time. The causal influences of time-variant factors on user satisfactions are time-variant. As shown in Figure 1(b), we consider the example of a user driven by popularity. If there is no recommendation at time t_1, the user would prefer news A (denoted by blue strip in Figure 1) about NVIDIA stocks at time t_2, possibly because he or she is a stakeholder of NVIDIA. However, if news B (denoted by orange strip) about the United States chip export controls is presented to the user at time t_1, he or she might sell his/her NVIDIA stock by time t_2 and thus lose interest in news A (denoted by blue strip), even if it is popular then. Conversely, if news C (denoted by green strip) about United States policy is shown to and clicked

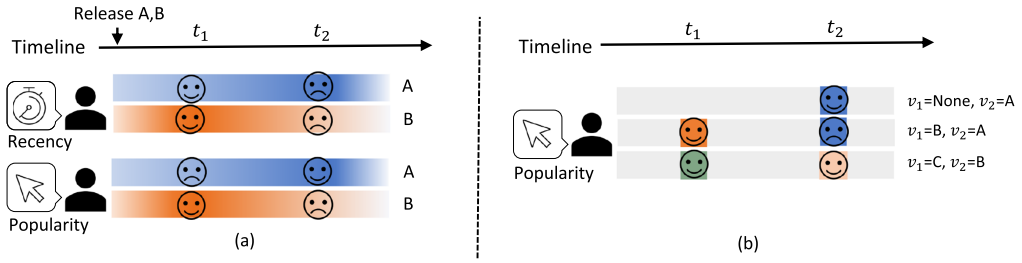


Fig. 1. Toy examples of (a) recency- and popularity-driven users, where positions further to the left indicate higher recency, and darker positions indicate higher popularity; (b) how the causal influence of exposed items affect future user satisfactions, where v_1, v_2 represents news items exposed at time t_1 and t_2, respectively.

by the user earlier, sparking their interest in policy, they would prefer news B (denoted by orange strip) about chip export controls at time t_2 despite its lack of popularity. A similar situation holds for recency-driven users. Therefore, the causal impact of time-variant factors on user satisfaction evolves over time, necessitating the modeling of causal relationships in an evolving manner.

Effectively modeling the evolving causal effects of past exposures on future user satisfaction requires separately capturing time-variant and time-invariant covariates, as these two types of covariates provide different types of user preference signals while also introducing potential confounding effects. In this article, we contend that these two types of covariates should be modeled independently. *Time-variant covariates*, consisting of various time-variant factors, risk overemphasizing recent trends or highly popular content if integrated into user modeling based on correlations. Separately modeling time-variant covariates allows for more precise modeling of the causal relationships among these time-variant factors and their confounding effects on the treated news and outcoming user satisfactions. Similarly, by isolating *time-invariant covariates*, recommenders can more accurately represent inherent user preferences independent of time, mitigating the impact of fluctuating noise and confounding effects arising from redundant associations. Therefore, separately modeling covariates in news recommendations from time-variant and time-invariant perspectives enables a more nuanced handling of confounding effects.

In this work, we approach the news recommendation task from a causal perspective, carefully addressing both time-variant and time-invariant covariates to accurately model the evolving causal influences of exposed news items on users' satisfaction. Previous sequential recommenders effectively captured information from sequences of historically interacted items. However, these models primarily rely on correlations between time-variant factors and interacted items, rather than explicitly learning the causal relationships between them over time, separate from time-invariant factors. To explicitly model the evolving causal relationships of time-variant covariates, we utilize a transformer-based causal block to incorporate news popularity and recency. Simultaneously, we extract users' inherent preferences from extensive textual information using **Large Language Models (LLMs)**, treating these preferences as the positive component of time-invariant covariates. In general, we design a ***CAuSal Time-aware Recommender (CAST-Rec)***, aiming to make recommendations based on the estimation of user satisfaction by modeling the evolving causal influences of news exposures. Although our primary focus is on news popularity and recency, CAST-Rec is adaptable and can seamlessly incorporate other time-variant factors that may be more prominent in specific scenarios.

We conclude our contributions as follows:

—To understand the causal influence of recommendations on user satisfaction over time, we are the first to consider the evolving causal influence of news exposures.

—To effectively model complex dependencies among time-variant factors, we design a transformer-based causal block to address the time-variant covariates.
—To carefully capture time-invariant covariates, we utilize the semantic understanding, generalization, and generation capabilities of LLMs.
—To validate the effectiveness of our proposed CAST-Rec, we conduct extensive experiments on two real-world news recommendation datasets.

2 Task Formulation

We denote a user $u \in \mathcal{U}$ and a news item $v \in \mathcal{V}$, where $\mathcal{U}$ and $\mathcal{V}$ are the sets of users and items, respectively. In this article, we re-scrutinize the task of news recommendations from a causal view. For each user u, the exposed item and the corresponding user satisfaction are treated as treatment and outcomes, denoted as variable $\mathbf{V}$ and variable $\mathbf{S}$, respectively. To represent the evolution of time, we define a set of time slices $\mathcal{T} = \{\mathbf{T}_i \in \mathbb{R}\}$, the past time period is denoted as $\mathbf{T}_{1:t} = \{\mathbf{T}_1, \ldots, \mathbf{T}_t\}$, and the prediction time is denoted as $\mathbf{T}_{t+1}$. In this work, we use subscripts to denote time. For example, the news item exposed at time $\mathbf{T}_i$ is represented as v_i, the sequence of news item exposed in the past is denoted as $\mathbf{V}_{1:t}$. To model the evolving causal influence of $\mathbf{V}$ on $\mathbf{S}$, we consider both time-variant covariates $\mathbf{C}$ and time-invariant covariates $\bar{\mathbf{C}}$. For time-variant covariates $\mathbf{C}$, c_i generally refers to the value of any time-variant covariate at time $\mathbf{T}_i$. We incorporate two time-variant factors as time-variant covariates: news popularity $\mathbf{P}$ and recency $\mathbf{R}$. Popularity and recency during the past time period $\mathbf{T}_{1:t}$ are indicated as $\mathbf{P}_{1:t} = \{p_i^v | v \in \mathbf{V}_{1:t}, 1 \le i \le t\}$ and $\mathbf{R}_{1:t} = \{r_i^v | v \in \mathbf{V}_{1:t}, 1 \le i \le t\}$, where p_i^v denotes the popularity of the exposed news item v at time $\mathbf{T}_i$, r_i^v denotes the corresponding recency. For time-invariant covariates $\bar{\mathbf{C}}$, we learn $\bar{c}$ for the user u from the user's long-term browsing history and/or demographic information.

2.1 News Recommendation from a Causal View

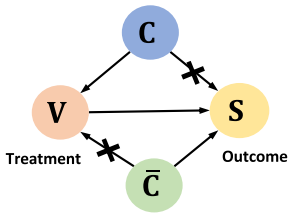
News recommendation methods predict users' preferences on candidate news items by analyzing their browsing history, where time is crucial. To capture the evolving causal relationship between exposed news items and user satisfaction while accounting for time, we consider both the causal influences of time-variant covariates and time-invariant covariates.

2.1.1 Covariates and Confounders. To clearly articulate our causal model, we first introduce covariates and confounders, with a particular focus on their relationships and distinctions.

A *covariate* is an endogenous variable that is observed in the scope of causal study and correlated with both the treatment variable and/or the outcome variable. In a causal study, covariates can serve as confounders, predictors, mediators, or nuisance variables, but only confounders directly affect the validity of causal inference. Covariates do not always introduce bias but provide additional information and can therefore be included in predictions.

A *confounder* is a specific type of covariate that simultaneously influences both the treatment and the outcome variables while not affecting the causal pathway between them as mediator. In causal inference, a confounder introduces bias into the estimation of causal effects because it builds spurious associations between the treatment and outcome variables. Therefore, additional techniques, such as backdoor adjustment or stratification, are required to carefully address confounders and ensure accurate causal estimation.

Previous studies have either solely addressed confounding effects [3, 4] or treated temporal factors as additional information [8, 9, 22, 31]. In this article, we consider both the additional information and the confounding effects introduced by time-variant covariates $\mathbf{C}$ and time-invariant covariates $\bar{\mathbf{C}}$. Taking the recency of browsed news $\mathbf{R}$ as an example time-variant covariate, it can improve



V: exposed news v_{t+1}
S: user satisfaction s_{t+1}
C: time-variant covariates **C**, e.g. popularity $\mathbf{P}_{1:t}$, recency $\mathbf{R}_{1:t}$.
$\bar{\mathbf{C}}$: time-invariant covariates $\bar{\mathbf{C}}$, e.g. user inherent preference.

Fig. 2. A structural causal model illustrating the *confounding effects* of covariates in news recommendations. Exposed news item v_{t+1} at prediction time $\mathbf{T}_{t+1}$ is treated as the treatment and user's satisfaction s_{t+1} on the item is treated as the outcome. This causal model considers both the confounding effects sourced from the time-variant covariates C for the past time period $\mathbf{T}_{1:t}$ and the confounding effects sourced from the time-invariant covariates $\bar{\mathbf{C}}$. As indicated by the crosses, we address the confounding effects from the time-variant covariate C and the time-invariant covariate $\bar{\mathbf{C}}$ via mitigating the information flow of paths $\mathbf{C} \rightarrow \mathbf{S}$ and $\bar{\mathbf{C}} \rightarrow \mathbf{V}$, respectively.

the system's accuracy by capturing nuanced user preferences related to the additional temporal information. However, recency **R** also introduce confounding effects by simultaneously influencing item exposures **V** and users' satisfaction **S**, as users may remain continuously focused on recently occurring news events.

2.1.2 Structural Causal Model. To address the confounding effects from both time-variant and time-invariant covariates, we depict the causal relationships in the scenario of news recommendations with a structural causal model. In the structural causal model depicted in Figure 2, the exposed news v_{t+1} at time $\mathbf{T}_{t+1}$ is considered as the treatment variable, while the corresponding user satisfaction **S** as the outcome representing how much the user likes the exposed news. Figure 2 also represents the two types of confounding effects sourced from time-variant covariates **C** and time-invariant covariates $\bar{\mathbf{C}}$, respectively. Directed arrows in Figure 2 depict the information flow from the cause factor to the effect factor, indicating that the child node is influenced by the ancestor node. Each edge is depicted as follows:

- $\mathbf{V} \rightarrow \mathbf{S}$ shows user satisfaction with the news item recommended by the system.
- $\mathbf{C} \rightarrow \mathbf{V}$ shows how time-variant covariates, such as the popularity and recency of correlated news items, influence the next news exposed to the user.
- $\mathbf{C} \rightarrow \mathbf{S}$ indicates the evolving nature of user preferences, showing how time-variant covariates influence user satisfaction with exposed items.
- $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ indicates user's time-invariant preferences which can be learned from the user's browsing history and/or demographic profile, will implicitly influence whether the news item will be exposed to the user by the system.
- $\bar{\mathbf{C}} \rightarrow \mathbf{S}$ shows that the user's time-invariant preferences which can be learned from browsing history and/or demographic profile affects their satisfaction toward the exposed news.

From a causal view, we aim at calculating the next recommended news item by predicting user satisfaction via estimating the causal influence of path $\mathbf{V} \rightarrow \mathbf{S}$. To this end, we have to carefully address the two types of confounding effects introduced by time-variant and time-invariant confounders to effectively leverage the benefits of these covariates while mitigating the drawbacks.

2.2 Confounding Effects on Recommendation

In Figure 2, we categorize the confounding effects of these three variables into two types: time-variant and time-invariant. This section will elaborate on how we account for and address these two types of confounder and their adverse effects on recommendations.

2.2.1 Time-variant Confounding Effect. In this article, we consider two time-variant covariates: news popularity $\mathbf{P}$ and recency $\mathbf{R}$. The popularity and recency change over time and can be derived from covariates in past user interactions. For example, the recency of a news v browsed at time $\mathbf{T}_i$ can be calculated as $\mathbf{T}_i - \mathbf{T}_v$, where $\mathbf{T}_v$ denotes the release time of the news article v or the first time the news v appears in the system, $\mathbf{T}_v \leq \mathbf{T}_i$. The time-variant covariates $\mathbf{C} = \{\mathbf{P}, \mathbf{R}\}$ concurrently influence both the exposed news treatment $\mathbf{V}$ and the outcome user satisfaction $\mathbf{S}$. Specifically, at time $\mathbf{T}_i \in \mathbf{T}_{1:t}$, substituting $\mathbf{C}$ by $\mathbf{P}$, the path $\mathbf{C} \rightarrow \mathbf{V}$ in Figure 2 indicates that when the system selects exposed news v, it considers the news popularity p_i^v at time $\mathbf{T}_i$; the path $\mathbf{C} \rightarrow \mathbf{S}$ shows that users' satisfaction for this news v can be influenced by its current popularity p_i^v, where generally popular news is more likely to gain favor. A similar situation holds for the recency variable $\mathbf{R}$.

The confounding effects of time-variant covariates $\mathbf{C}$ negatively influence the estimation of the causal relationship between $\mathbf{V}$ and $\mathbf{S}$ because it provides a backdoor path $\mathbf{V} \leftarrow \mathbf{C} \rightarrow \mathbf{S}$. Treating item popularity $\mathbf{P}$ as a time-variant covariates, this confounding effect is also known as popularity bias in recommender systems. For example, political news is rarely viewed by the youth, who are more active on the news platform compared to the middle-aged demographic. However, the middle-aged, who have an interest in political news, may not be exposed to these news items if the system is largely influenced by popularity. In this article, we initially apply time-aware stratification to separate these time-variant covariates, i.e., news popularity $\mathbf{P}$ and recency $\mathbf{P}$, from the interacted news items, then address the confounding by blocking the backdoor path $\mathbf{C} \rightarrow \mathbf{S}$ using the CAST-Rec framework. The rationale of time-aware stratification will be detailed in Section 3.1, and our proposed CAST-Rec will be introduced in Section 3.3.

2.2.2 Time-invariant Confounding Effect. As for the time-invariant covariate $\bar{\mathbf{C}}$, we account for the news content extracted from browsing history or user demographic profiles. Although these time-invariant features bring benefits to the prediction, they also bring confounding effects by concurrently influencing both the exposed news treatment $\mathbf{V}$ and the outcome user satisfaction $\mathbf{S}$, as shown in Figure 2. Specifically, the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ illustrates that the system's exposures implicitly mirrors time-invariant user inherent preferences; concurrently, these time-invariant preferences $\bar{\mathbf{C}}$ also explicitly reflected in the satisfactions $\mathbf{S}$, represented by path $\mathbf{F} \rightarrow \mathbf{S}$.

Similarly, these time-invariant features from either browsing history or demographic profile $\bar{\mathbf{C}}$ introduce confounding effects through a backdoor path $\mathbf{V} \leftarrow \bar{\mathbf{C}} \rightarrow \mathbf{S}$. To carefully leverage the time-invariant covariate, we extract and incorporate the textual information from extensive browsing sequences into our predictions, while blocking the implicit influences of the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ to circumvent the confounding. In this article, the proposed CAST-Rec extracts and models the information flow through the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$ with LLMs, as elaborated in Section 3.3.

2.3 Problem Formulation

Based on the structural causal model depicted in Figure 2, we formulate the problem we focus in our proposed CAST-Rec for news recommendations as below.

Problem 1 (News Recommendations). *For each user u, we aim to learn a news recommendation model g that is capable of predicting user satisfaction score $\mathbf{s}_{t+1}$ based on estimating the causal effects of exposed news items v_{t+1} at prediction time $\mathbf{T}_{t+1}$:*

$$\mathbf{s}_{t+1} = g(v_{t+1} | \bar{\mathbf{C}},\ \mathbf{V}_{1:t},\ \mathbf{P}_{1:t},\ \mathbf{R}_{1:t}), \tag{1}$$

where $\mathbf{V}_{1:t}$ includes the historical browsed news items, $\mathbf{P}_{1:t}$ represents the corresponding popularity, and $\mathbf{R}_{1:t}$ involves the recencies during the past time period $\mathbf{T}_{1:t}$. The extracted preferences of user u acting as time-invariant covariate are represented as $\bar{\mathbf{C}}$.

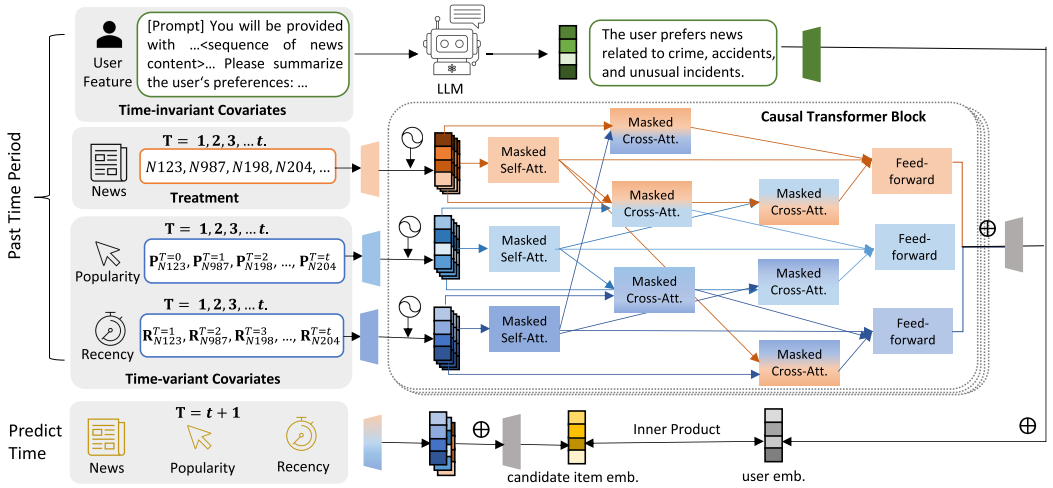


Fig. 3. Overview of our proposed CAST-Rec framework.

3 Methodology

We propose CAST-Rec to predict user's satisfactions on exposed news by estimating the evolving causal relationship between them, taking into account both time-variant and time-invariant covariates. Before learning the evolving influences, we first prove the rationality of independently modeling the exposed news and the time-variant covariates (i.e., news popularity and recency) in Section 3.1. Then, as shown in Figure 3, CAST-Rec carefully leverage the information sourced from time-invariant covariates with LLMs (Section 3.2) and time-variant covariates with a series of transformer-based causal blocks (Section 3.3), respectively.

3.1 Causality Preparation: Time-aware Stratification

Before learning, we carefully investigate the rationality of independently modeling the treatments variable (i.e., exposed news) and the time-variant covariates (i.e., time-variant factors such as recency and popularity). Inspired by previous works on causality [2, 21], we employ a commonly used stratification method to retrieve the independence between the treatment news $\mathbf{V}$ and time-variant covariates $\mathbf{C}$ in Figure 2, allowing for separate modeling of the covariates and the treatment. This section first introduces the rationale of stratification method in causality, then theoretically prove the independence of $\mathbf{C} \perp\!\!\!\perp \mathbf{V}$ based on the structural causal model in Figure 2.

3.1.1 Stratification in Causality. The idea of stratification is to divide the data into two or more distinct strata (i.e., subgroups) such that the confounder $\mathbf{C}$ and $\mathbf{V}$ are independent of each other in each subgroup. A key challenge of applying stratification is how to divide users to ensure that within each subgroup, the independence $\mathbf{C} \perp\!\!\!\perp \mathbf{V}$ always exists. To resolve this, we divide subgroups by numerous time slices, the time intervals of which tend to be infinitesimal. Within each subgroup with time slices $\mathbf{T}_i$, the time-variant covariates can be regarded as static. Specifically, based on the casual model shown in Figure 2, the probability of the exposure of news item at time $\mathbf{T}_i$ can be presented as $P(\mathbf{V} = 1|\mathbf{C} = c_i, \bar{\mathbf{C}})$. At time $\mathbf{T}_i$, the user either be exposed with news item or not; only the satisfaction $\mathbf{S}$ of the users who are exposed to the news is taken into account. However, for the recommendations in a past time slices $\mathbf{T}_i \in \mathbf{T}_{1:t}$, the partition of users who are exposed with the news item is determined, yielding the independence of $\mathbf{C} \perp\!\!\!\perp \mathbf{V}$. The following subsections proves

the effectiveness of this time-aware stratification. For simplicity, we omit v and the time-invariant variable $\bar{\mathbf{C}}$ for the probability $P(\mathbf{V}=1|c_i,\bar{\mathbf{C}})$.

3.1.2 Proof of Independence. We aim to prove the independence within each subgroup that $\mathbf{T}_i \in \mathbf{T}_{1:t}$ in order to separately model the time-variant covariate variables (i.e., popularity and recency of items). With any subgroup $\mathbf{T}_i$ and the corresponding time-variant covariate $\mathbf{C}=c_i$, each user u has either been exposed to the news v (divided into partition $u \in \mathcal{U}_1$) or not being exposed to the news v (divided into partition $u \in \mathcal{U}_2$). For simplicity, we omitted the notation v of news. Formally, the two partitions of users can be presented as:

$$\begin{aligned} \mathcal{U}_1 &= \{u \in \mathcal{U} | P(\mathbf{V}=1|c_i,\ u) = 1 = P(\mathbf{V}=1|u)\} \text{ and} \\ \mathcal{U}_2 &= \{u \in \mathcal{U} | P(\mathbf{V}=0|c_i,\ u) = 1 = P(\mathbf{V}=1|u)\}, \\ \mathcal{U}_1 &\cup \mathcal{U}_2 = \mathcal{U} \end{aligned} \tag{2}$$

where $P(\mathbf{V}=1|c_i,u)$ represents the probability of news item $\mathbf{V}$ has been exposed to user u given the time-variant covariate $\mathbf{C}=c_i$. Intuitively, the user variable $\mathbf{U}$ is independent of all the variables in Figure 2. Given the causal graph in Figure 2, within the subgroup of $\mathbf{T}_i$, the probability of $\mathbf{V}=1$ can be represented as below:

$$\begin{aligned} &P(\mathbf{V}=1|\mathbf{C}=c_i) \\ &\overset{(a)}{=\!=} \sum_{u\in\mathcal{U}} P(\mathbf{V}=1|\mathbf{C}=c_i,\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\ &\overset{(b)}{=\!=} \sum_{u\in\mathcal{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\ &+ \sum_{u\in\mathcal{U}_2} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\ &\overset{(c)}{=\!=} \sum_{u\in\mathcal{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u|\mathbf{C}=c_i) \\ &\overset{(d)}{=\!=} \sum_{u\in\mathcal{U}_1} P(\mathbf{V}=1|\mathbf{U}=u)P(\mathbf{U}=u) = P(\mathbf{V}=1), \end{aligned}$$

where Equation (a) uses the law of total probability by conditioning on all possible users $\mathbf{U}$, Equations (b) and (c) are based on Equation (2), and Equation (d) is based on the independence $\mathbf{U} \perp\!\!\!\perp \mathbf{C}$. Therefore, the time-variant covariate $\mathbf{C}$ is independent of the news exposure $\mathbf{V}$. Given $\mathbf{C} \perp\!\!\!\perp \mathbf{V} \mid \mathbf{U}$, we are able to separately model the covariate $\mathbf{C}$ and exposed items $\mathbf{V}$. In this article, we account for both item popularity and recency for time-variant covariates, $\mathbf{C}=\{\mathbf{P},\mathbf{R}\}$. A detailed illustration of how we leverage these time-variant covariates will be given in Section 3.3.

3.2 LLM for Time-invariant Covariates

Considering the confounding effects of backdoor path $\mathbf{V} \leftarrow \bar{\mathbf{C}} \rightarrow \mathbf{S}$, CAST-Rec utilizes LLMs to extract the essential information flow of $\bar{\mathbf{C}} \rightarrow \mathbf{S}$ from historical browsing sequences, while mitigating the path $\bar{\mathbf{C}} \rightarrow \mathbf{V}$. This is motivated by the following three key capabilities of LLMs: (1) The powerful *semantics understanding capability* of LLMs empowers them to uncover nuanced user interests by interpreting deeper meanings embedded within content. For instance, when analyzing news articles, LLMs can go beyond understanding the explicit semantic meaning to capturing subtle, implicit emotions, such as the humor or hilarity associated with mentioning a specific comedian's name. This ability to detect both semantics information and hidden emotional cues enables LLMs

to deliver more personalized and contextually aware recommendations. Traditional methods, such as using language models like BERT, may overlook these nuances and fail to capture inherent user preferences effectively during user modeling. (2) The *generalization capability* of LLMs is harnessed to abstract away minor variations, generating a meaningful time-invariant representation based on users' historical interactions. For example, while the content of a news article remains static, underlying elements like writing style may subtly evolve over time. These changes, although time-varying, are too subtle to justify modeling explicitly as time-variant variables. In contrast, traditional approaches like content-aware attention may inadvertently incorporate these subtle time-varying features, leading to potential noise and reduced clarity in the resulting representations. (3) The *generative capability* of LLMs enhances transparency by producing human-readable summaries of previously browsed news. By distilling complex data or lengthy content into concise summaries, LLMs provide users with a clearer understanding of the underlying information, making it more accessible and actionable. Specifically, these summaries allow users to directly observe how the system models their inherent interests in news articles, independent of temporal factors such as writing style or popularity. In contrast, traditional methods including attention mechanisms often lack the ability to explicitly express how user interests are represented, making their decision-making process less interpretable and transparent.

The employment of LLMs here contributes in a two-fold manner: (1) *mitigating the path* $\bar{\mathbf{C}} \rightarrow \mathbf{V}$: The time-varying nature of the historical browsing sequence $\mathbf{V}_{1:t}$ arises from its ordering based on interaction time. Therefore, directly encoding this sequence as a time-invariant covariate $\bar{\mathbf{C}}$ using language models is inaccurate. However, when browsing news items are shuffled, the advanced generative capabilities of LLMs allow them to identify and isolate inherent preferences that remain consistent over time. (2) *Enhancing the path* $\bar{\mathbf{C}} \rightarrow \mathbf{S}$: The purpose of encoding historical browsing contents in news recommendations is to capture inherent user preference from the textual data [22]. LLMs are capable of understanding complex language patterns and semantics from vast amounts of textual information, allowing them to provide highly contextualized and concise summaries. These diverse yet concise summaries generated by the LLMs helps capture the nuances of user preferences, thereby enhancing the modeling of $\bar{\mathbf{C}} \rightarrow \mathbf{S}$.

Specifically, to capture user time-invariant inherent preferences from the browsing history, there are intuitively two approaches: (1) encoding the news content with language models first and then concatenating it as a sequence, or (2) concatenating first and then encoding. This approach aims to incorporate the user's interactions with various news items over time, providing a comprehensive view of their interests. The modeling of sequential information in the first method heavily depends on the performance of language models and can harm the semantic coherence between two consecutively browsed news items. Therefore, we adopt the second approach, first concatenating the textual feature f^v (e.g., title, abstract, category of news), of each news item v in chronological order. In CAST-Rec, we adopt the title for its balance between being informative and concise. The textual information of news features browsed over past time period $\mathbf{T}_{1:t}$ can be represented as $f^{v_1} \,\|\, f^{v_2} \,\|\, \ldots \,\|\, f^{v_t}$, where v_t represents the news item interacted with at time t. Furthermore, given that an extensive browsing history for active users introduces redundant relationships, we aim for user interests to be concise and consistent over time. The concatenation $f^{v_1} \,\|\, f^{v_2} \,\|\, \ldots \,\|\, f^{v_t}$ of news contents is formatted with a prompting template $Template(\cdot)$ as shown in Figure 4, which instructs the LLM to summarize the essence of user interests with one sentence. This process yields a formatted personalized history $Prompt = Template(f^{v_1} \,\|\, f^{v_2} \,\|\, \ldots \,\|\, f^{v_t})$. By providing a clear and structured prompt, the LLM can effectively generate concise and relevant summaries that capture the essential aspects of the user's interests. The box in the upper left corner of Figure 3 shows an example of a prompt. Then, this personalized history *Prompts* is tokenized into the input sequence of tokens $X = (x_1, x_2, x_3, \ldots, x_m)$, where m is the length of input sequence. The LLM

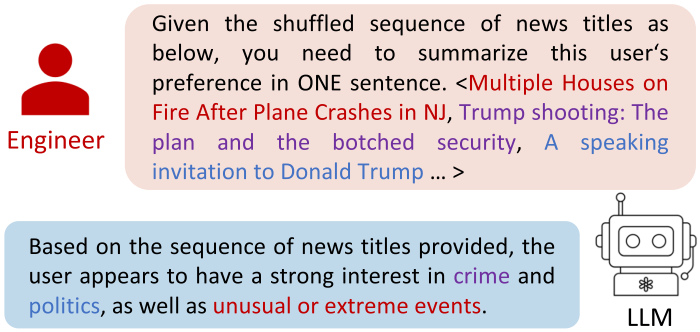


Fig. 4. An example prompt template used in the chat with the LLM.

generates an output sequence, we take the last hidden states of this generation as the representation of time-invariant user interest $\bar{c}$. For a LLM with a number of layer denoted by $L(\cdot)$, we can have the time-invariant user representation as follows:

$$\mathbf{e}^{\bar{C}} = \mathbf{H}_{LLM}^{(l)} = L^{(l)}(h_1^{(l-1)}, h_2^{(l-1)}, ..., h_m^{(l-1)}) \tag{3}$$

where $\mathbf{H}_{LLM}^{(l)}$ represents the hidden states of the LLM at the last layer, $h_i^{(l-1)} \in \mathbb{R}^{\frac{d_1}{n}}$ is the embedding of ith token at the l-1th layer, and $\bar{c} \in \mathbb{R}^{d_1}$ is the representation of user interest at the last layer, d_1 is the dimensionality of the output from the LLM model.

This time-invariant user interest $\bar{c}$ can be decoded into an explicit one-sentence user interest summarized from the browsing history. Since only news titles are used along with the prompt instructions, and without inputting any temporal information, the output $\bar{c}$ is expected to capture the essential user interests, which are consistent over time.

3.3 Time-aware Modeling for Time-variant Covariates

Based on the time stratification preparation in Section 3.1, we are capable of separately modeling the treated exposed news and potential time-variant covariates. This section consider two time-variant covariates, i.e., news popularity $p \in \mathbf{P}$ and recency $r \in \mathbf{R}$, and conduct time-aware modeling to carefully leverage these two time-variant covariates.

3.3.1 Popularity. The **Click-Through Rate (CTR)** of a news item can be calculated at four different granularities: news category, news subcategory, the news item itself, and words in the news title. For example, the CTR based on news category represents the probability of a click occurring on news items within the same category as the exposed item v, given that news items in the same category are exposed:

$$p_i^v = \frac{1}{|\mathcal{U}_1^{v'}||\mathcal{V}^{cat_v}|} \sum_{u \in \mathcal{U}_1^{v'}} \sum_{v' \in \mathcal{V}^{cat_v}} P(\mathbf{S} = 1 | u, v', \mathbf{T}_i), \tag{4}$$

where $u \in \mathcal{U}_1^{v'} \subset \mathcal{U}_1$ represents the set of users exposed to news item v', $v' \in \mathcal{V}^{cat_v}$ represents the set of news items in the same category of the exposed item v, and the conditional probability $P(\mathbf{S} = 1 | u, v', \mathbf{T}_i)$ represents the probability of user u clicks given the exposure of item v' at time $\mathbf{T}_i$.

3.3.2 Recency. The recency of news is calculated by the reciprocal of the difference between current time t and the time that it was first exposed in the system. Similar to the popularity, the recency also experimented with the four different granularities. Formally, the category-level

recency of news item v at time $\mathbf{T}_i$ can be represented as follows, denoted by $r_i^v \in \mathbb{R}$:

$$r_i^v = min(0,\ min_{v' \in \mathcal{V}^{cat_v}}(\mathbf{T}_i - \mathbf{T}_{v'})), \tag{5}$$

where $\mathbf{T}_{v'}$ denotes the first time that news $v' \in \mathcal{V}^{cat_v}$ is exposed to any user $u \in \mathcal{U}$ in the system.

3.3.3 Multi-input Causal Block. As time-variant covariates, the news popularity and recency provide invaluable information for making predictions. To this end, we employ a transformer-based multi-input causal block to model the evolving causal influence between these covariates and exposed news items, aiming to improve the prediction of user satisfaction.

Inputs. For each past time slice $\mathbf{T}_i \in \mathbf{T}_{1:t}$, three time-variant variables are treated as the inputs: (i) treatment variable exposed news v_i, corresponding (ii) covariate variable news popularity p_i, and (iii) covariate variable news recency r_i. For simplicity, we omit the notation of v for the popularity and recency here. Given the differing dimensions of these three time-variant variables, each input is projected through its respective multi-perception layer before being processed by the causal block layers. Note that the embeddings for exposed news are initialized using Xavier uniform initialization. For instance, exposed news items arranged in chronological sequence $\mathbf{V_{1:t}} = \{v_1, v_2, \ldots, v_t\}$ are formally projected as the following formulation:

$$\mathbf{H}_0^{\mathrm{V}} = MLP^{\mathrm{V}}(v_1 \,\|\, v_2 \,\|\, \ldots \,\|\, v_t), \tag{6}$$

where $\mathbf{H}_0^{\mathrm{V}} \in \mathbb{R}^{d_2 \times t}$ is the hidden representation of the series of treatments at time $\mathbf{T}_{1:t}$, d_2 is the dimensionality of hidden representation in CAST-Rec. The projection network is a multi-layer perceptron MLP^{V} consisting of two linear layers with the Tanh activation function in between. Similarly, the news popularity $\mathbf{P}_{1:t} \in \mathbb{R}^t$ and the news recency $\mathbf{R}_{1:t} \in \mathbb{R}^t$ are projected using MLP^{P} and MLP^{R}, respectively, yielding $\mathbf{H}_0^{\mathrm{P}}$ and $\mathbf{H}_0^{\mathrm{R}}$. Both MLP^{P} and MLP^{R} consist of two linear layers with an ELU activation function in between. This difference arises because the ELU activation function is particularly effective for positive inputs. This is the case for popularity and recency as formulated in Equations (4) and (5). In contrast, the Tanh activation function helps to normalize the embedding of exposed news.

Self-/Cross-attention. The initially hidden representations $\mathbf{H}_0^{\mathrm{V}}, \mathbf{H}_0^{\mathrm{P}}, \mathbf{H}_0^{\mathrm{R}}$ for exposed news, popularity, and recency are then inputted into the transformer-based causal block. Specifically, they are first fed into multiple attention heads, each requiring a set of query, key, and value vectors, denoted as $Q^{(n)}, K^{(n)}, V^{(n)} \in \mathbb{R}^{T \times \frac{d_2}{N}}$, respectively. The attention heads are indexed by n, and there are N such heads in total. For any initial hidden representation $\mathbf{H}_0 \in \{\mathbf{H}_0^{\mathrm{V}}, \mathbf{H}_0^{\mathrm{P}}, \mathbf{H}_0^{\mathrm{R}}\}$:

$$\begin{aligned} Q^{(n)} &= \mathbf{H}_0 W_Q^{(n)} + \mathbf{1} b_Q^{(n)\top} \\ K^{(n)} &= \mathbf{H}_0 W_K^{(n)} + \mathbf{1} b_K^{(n)\top}, \\ V^{(n)} &= \mathbf{H}_0 W_V^{(n)} + \mathbf{1} b_V^{(n)\top} \end{aligned} \tag{7}$$

where $W_Q^{(n)}, W_K^{(n)}, W_V^{(n)} \in \mathbb{R}^{d_2 \times \frac{d_2}{N}}$ represents weights for the ith self-attention head, $b_Q^{(n)}, b_K^{(n)}, b_V^{(n)} \in \mathbb{R}^{\frac{d_2}{N}}$ are the biases for query, key, values, respectively, and $\mathbf{1} \in \mathbb{R}^{\frac{d_2}{N}}$ is the vector of ones. The attention values in the ith head are formally computed as follows:

$$Attn^{(n)}(Q^{(n)}, K^{(n)}, V^{(n)}) = softmax\left(\frac{Q^{(n)} K^{(n)\top}}{\sqrt{\frac{d}{N}}}\right) V^{(n)}. \tag{8}$$

The outputs of these N attention heads are then concatenated:

$$MHA(Q, K, V) = (Attn^{(0)}, ..., Attn^{(n)}, ..., Attn^{(N)}). \tag{9}$$

We first compute the self-attention representation $\dot{\mathbf{H}}$. By substituting the hidden representation $\mathbf{H}_0$ in Equation (7) with $\mathbf{H}_0^{\mathrm{V}}$, we can calculate the respective query, key, and values $Q^{\mathrm{V}}, K^{\mathrm{V}}, V^{\mathrm{V}}$. For simplicity, we omitted the index i for multiple heads. Consequently, the self-attention representation $\dot{\mathbf{H}}^{\mathrm{V},(l)}$ at the lth causal block is computed as follows:

$$\dot{\mathbf{H}}^{\mathrm{V},(l)} = SelfAtt(\mathbf{H}^{\mathrm{V}}), \tag{10}$$

$$SelfAtt(\mathbf{H}^{\mathrm{V}}) = LN(FF(MHA^{(l)}(Q^{\mathrm{V}}, K^{\mathrm{V}}, V^{\mathrm{V}})) + \dot{\mathbf{H}}^{\mathrm{V},(l-1)}). \tag{11}$$

Here, $MHA^{(l)}$ represents the multi-head attention in Equation (9) in the lth blocks, the feed-forward network is denotes as $FF(\cdot) = Linear(ReLU(Linear(\cdot)))$, and $LN(\cdot)$ stands for linear normalization. To avoid over-fitting, the final linear projection layer present in the original transformer decoder [32] is omitted. Similarly, the variables $\mathbf{P}$ and $\mathbf{R}$ of news popularity and recency are also self-attentively embedded into $\dot{\mathbf{H}}^{\mathrm{P},(l)}$ and $\dot{\mathbf{H}}^{\mathrm{R},(l)}$, respectively. Calculating self-attention allows the model to better understand intra-variable evolving dependencies.

Next, we compute the pairwise cross-attentions $\tilde{\mathbf{H}}$ to capture inter-variable evolving dependencies. Given self-attention outputs $\dot{\mathbf{H}}$ from Equation (11), the cross-attention can be calculated from both directions: taking the cross-attention between $\mathbf{V}$ and $\mathbf{P}$ as an example, using $\dot{\mathbf{H}}^{\mathrm{V}}$ as the query and $\dot{\mathbf{H}}^{\mathrm{P}}$ as the key and values, and vice versa. Calculating cross-attention in both directions allows the models to fully capture the evolving dependencies over time. Specifically, beyond the causal influence represented by $\mathbf{V}_i \rightarrow \mathbf{P}_i$ at time $\mathbf{T}_i$, the exposure v_i will also influences the news popularity at time $p_{i+1}^{v_i}$, denoted by $\mathbf{V}_i \rightarrow \mathbf{P}_{i+1}$:

$$\begin{aligned} \tilde{\mathbf{H}}^{\mathrm{V,P}(l)} &= CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}}) \\ \tilde{\mathbf{H}}^{\mathrm{P,V}(l)} &= CrossAtt(\dot{\mathbf{H}}^{\mathrm{P}}, \dot{\mathbf{H}}^{\mathrm{V}})^{,} \end{aligned} \tag{12}$$

$$CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}}) = LN(FF(MHA^{(l)}(Q^{\mathrm{V}}, K^{\mathrm{P}}, V^{\mathrm{P}})) + \mathbf{H}^{\mathrm{V}}), \tag{13}$$

where $CrossAtt(\dot{\mathbf{H}}^{\mathrm{P}}, \dot{\mathbf{H}}^{\mathrm{V}})$ is calculated in the similar way as $CrossAtt(\dot{\mathbf{H}}^{\mathrm{V}}, \dot{\mathbf{H}}^{\mathrm{P}})$. This process helps the model understand how the covariates influence the exposure of news items over time. By learning these dependencies, the model can make more accurate predictions based on the combined effects of news exposures $\mathbf{V}$, popularity $\mathbf{P}$, and recency $\mathbf{R}$.

After computing the cross-attention, a feed-forward layer is adopted to combine all the relative information of a variable, for example,

$$\mathbf{H}^{\mathrm{V}} = FF(\tilde{\mathbf{H}}^{\mathrm{V,P}}, \tilde{\mathbf{H}}^{\mathrm{P,V}}). \tag{14}$$

Finally, after L multi-input causal blocks, the final output will be self-attentive aggregated into a time-variant user representation:

$$\mathbf{e}^{\mathrm{C}} = SelfAtt(\mathbf{H}^{\mathrm{V}} \parallel \mathbf{H}^{\mathrm{P}} \parallel \mathbf{H}^{\mathrm{R}}). \tag{15}$$

Position Encoding. In the original transformer model [32], positional encoding is used to preserve the order of hidden states within a sequence, as the attention mechanism tends to overlook order information. This position encoding is crucial in news recommendations to ensure that the temporal sequence of news exposures is accurately captured.

In our multi-input causal block, we employ relative positional encoding. This choice is made because absolute time information is already captured in the representations of news popularity and recency, which are calculated based on timestamps. Relative positional encoding allows us to focus on the relative time distances between news items, which is essential for modeling the evolving influence of earlier news exposures on later interactions. Considering the news items at

position $\mathbf{T}_i$ and $\mathbf{T}_j$ for any $\mathbf{T}_i \leq \mathbf{T}_j$, we have the following modifications to the key and value in the attention calculation:

$$\begin{aligned} \alpha_{j,i}^K &= W_{i-j}^K \\ \alpha_{j,i}^V &= W_{i-j}^V \end{aligned}. \tag{16}$$

where W_{j-i}^K, and $W_{j-i}^V \in \mathbb{R}^{t \times \frac{d}{N}}$ are two relative positional encoding matrices applied to the keys and values in the attention mechanism, respectively. Then, the scores $\alpha_{j,i}^K$ and $\alpha_{j,i}^V$ are used to adjust the attention score $Attn(Q,K,V)_j$ from Equation (8):

$$\begin{aligned} Attn(Q,K,V)_j &= \sum_{i \leq j} softmax_i \left(\frac{Q_j K_i'}{\sqrt{\frac{d}{N}}} \right) V_i', \\ K_i' &= K_i + \alpha_{j,i}^V, \quad V_i' = V_i + \alpha_{j,i}^V \end{aligned} \tag{17}$$

where $softmax_i$ calculates relative to position $\mathbf{T}_j$. Note that the latter position $\mathbf{T}_j$ only participates in the attention calculation for past positions $\mathbf{T}_i$ or itself.

3.4 Recommendation

Last, CAST-Rec makes predictions by integrating the causal influences from the exposed items and the two covariates, via three information flows $\mathbf{C} \rightarrow \mathbf{S}$, $\mathbf{V} \rightarrow \mathbf{S}$, and $\bar{\mathbf{C}} \rightarrow \mathbf{S}$, respectively.

Objective. At prediction time $\mathbf{T}_{t+1}$, we calculate the representation of any candidate news item $\hat{v}$ for exposure based on its popularity $p_{t+1}^{\hat{v}}$, recency $r_{t+1}^{\hat{v}}$:

$$\mathbf{e}^{\hat{v}} = MLP(MLP^{\mathrm{V}}(\hat{v}) \parallel MLP^{\mathrm{P}}(p_{t+1}^{\hat{v}}) \parallel MLP^{\mathrm{R}}(r_{t+1}^{\hat{v}})), \tag{18}$$

where we use an $MLP(\cdot)$ that includes two linear layers with ELU activation function to project the candidate news item into the matching space. Then, we estimate the user satisfaction $\mathbf{s}_{t+1}$:

$$\mathbf{s}_{t+1} = (\mathbf{e}^{\bar{\mathrm{C}}} \parallel \mathbf{e}^{\mathrm{C}})^\top \cdot \mathbf{e}^{\hat{v}}. \tag{19}$$

We trained our CAST-Rec model using the **Bayesian Personalized Ranking (BPR)** loss [26], which aims to maximize the difference in the user satisfaction scores between positive and negative candidate items. Formally:

$$\mathcal{L}_{BPR} = -\frac{1}{|O|} \sum_{(v,v') \in O} ln\ \sigma(\mathbf{s}^v - \mathbf{s}^{v'}), \tag{20}$$

where $\mathbf{s}^v$ and $\mathbf{s}^{v'}$ are the predicted user satisfaction scores for news items v and v', respectively. $O = \{(v,v') | v \in O^+, v' \in O^-\}$ is the training data, O^+ involves the positive interacted news items, while O^- is the sampled negative interacted news items, $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

Interpretability. Regarding the interpretability of the recommendation process, CAST-Rec introduces the following advancements: First, it applies causal inference to uncover more interpretable variable relationships compared to correlation-based methods. Second, it accounts for confounding effects from both time-variant and time-invariant covariates, reducing spurious correlations that introduce bias and improving model transparency. Third, CAST-Rec leverages LLMs to generate content that explicitly reflects users' time-invariant interests, providing a clear view of their inherent preferences and enhancing user trust in the system.

Table 1. Statistical Information of Datasets

Datasets	#Users	#Items	#Interactions	Density
MIND-small	1,809	12,668	37,801	1.80×10^{-4}
MIND-large	110,780	26,313	1,305,190	0.69×10^{-4}

4 Experiments

To validate the effectiveness of our proposed CAST-Rec, we have conducted extensive experiments on two real-world news recommendation datasets to answer the following research questions:

— *RQ1*: How does our proposed CAST-Rec perform in the news recommendation task compared to **State-of-the-Art (SOTA)** news recommenders?
— *RQ2*: How do different LLMs impact the performance of CAST-Rec?
— *RQ3*: How do different hyperparameters in CAST-Rec affect its performance?
— *RQ4*: How efficient is CAST-Rec?

4.1 Experiment Settings

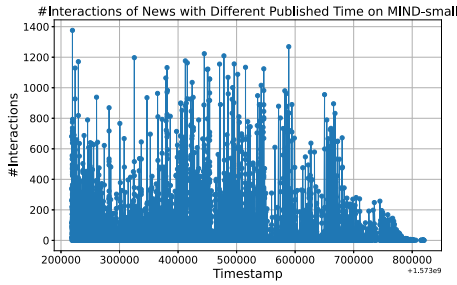
This section presents the datasets, baselines, evaluation protocol, and implementation details used in our experiments and comparisons to validate the effectiveness of CAST-Rec.

4.1.1 Datasets. We evaluate the effectiveness of our proposed CAST-Rec on the two public news recommendation datasets: MIND-small and MIND-large [49]. MIND dataset is collected from user behavior logs on the Microsoft News platform. The author randomly sampled 1 million active users with more than five news clicks on the platform during the period from 12 October 2019 to 22 November 2019. Similarly, the MIND-small version randomly samples 50,000 active users and their behavior logs. For each user, click behaviors before this time period are regarded as history, while exposed news and user interactions in this period are formatted into impressions. Different from the original MIND dataset, we organize the impressions into a series of accumulated sequences arranged chronologically by the timestamp of each impression. To exclude very short impression sequences [1, 25], we filtered out those with fewer than 10 behavior logs. Following common practices in the sequential recommendation, we compile the most recently clicked news for each user into the test set and the second most recently clicked news into the validation set.

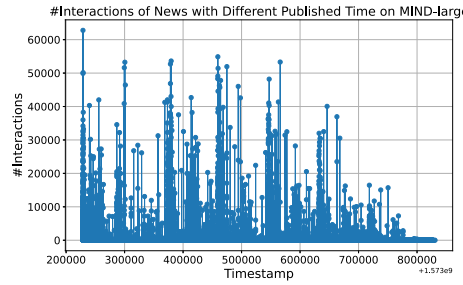
Furthermore, MIND contains over 160,000 English news articles, each featuring rich textual content, including the title, abstract, and category. The popularity of each news article is calculated based on its click-through rate within a recent time window [4, 22]. The recency of each news article is represented by the time elapsed since its initial exposure in the system, regardless of whether it was clicked [17, 22]. Details of MIND-small and MIND-large datasets used in our experiments are formatted in Table 1. The interaction distributions of news in these two datasets are visualized in Figure 5. It can be observed that there is a complex correlation between the publish time of news items and the corresponding interactions, indicating that relying solely on recency as the time-variant covariate is insufficient for modeling time-aware causal influences. Additionally, compared to MIND-small, users' interactions in the MIND-large dataset have higher variance.

4.1.2 Baselines. In our experiments, we have compared our proposed CAST-Rec with three categories of NRSs as follows.

Traditional NRSs represent user preferences based on news textual information while ignoring the evolving time. We selected four common news recommenders:



(a) User-item interaction distribution of news in `MIND-small`.



(b) User-item interaction distribution of news in `MIND-large`.

Fig. 5. Interaction distributions with respect to the interacted time (i.e., Timestamp)in the two datasets.

—DKN [34] utilizes a knowledge-aware convolutional neural network that integrates semantics and knowledge-level representations of news and an attention module to dynamically aggregate use history with current candidate news.
—NPA [44] integrates a CNN-based news representation model and a user representation model with word- and news-level attention mechanisms, enhanced by a personalized attention network using user ID embeddings to tailor the representation of news and users.
—NRMS [45] leverages multi-head self-attention in encoding the news titles and user browsing history, supplemented by additive attention to highlight important words in news content for more informative representations.
—NAML [43] unifies the representations of news titles, bodies, and categories from user browsing history with attention at both word- and user-interest levels.

Sequential NRSs incorporate sequential information from user browsing history in addition to news content. We include three models in this category:

—LSTUR [1] learns long-term user representations from user ID embeddings and encodes short-term user representations from history-browsed news contents.
—CAUM [23] incorporates a candidate-aware CNN network for modeling short-term user interests and a candidate-aware attention network for aggregating previously clicked news to build a comprehensive candidate-aware user representation.
—MINS [36] designs a GRU-based interest network to extract potential multiple interests of users from their historical browsed sequences.

Temporal NRSs consider temporal data into predictions, represented by the following two methods:

—PP-Rec [22] combines a personalized matching score based on user browsing history with a news popularity score that encodes click-through rates, recency, and news content.
—TCCM [4] analyzes the causal effects of time, news popularity, and the alignment score between news content and user interest in user behaviors, and it also estimates news popularity by learning from the granularity of entities and words.

4.1.3 Evaluation Protocol. The recommendation performance of our CAST-Rec is evaluated using the following metrics, all of which are commonly employed in similar tasks in [4, 14, 22, 44].

—**Mean Reciprocal Rank (MRR)** considers the rank position of the first target item in the ranked list. A higher MRR score indicates better performance.

Table 2. Recommendation Performance Comparison of CAST-Rec with Nine Baseline Models on the MIND-Small and the MIND-Large Datasets

Model		MIND-small				MIND-large			
		MRR	AUC	NDCG@5	NDCG@10	MRR	AUC	NDCG@5	NDCG@10
Traditional NRS	DKN	0.3092	0.6011	0.3312	0.3936	0.3217	0.6196	0.3479	0.4055
	NPA	0.3163	0.6329	0.3522	0.4001	0.3309	0.6542	0.3671	0.4197
	NRMS	0.3122	0.6407	0.3457	0.4066	0.3310	0.6602	0.3599	0.4201
	NAML	0.3451	0.6631	0.3699	0.4309	0.3620	0.6834	0.3823	0.4652
Sequential NRS	LSTUR	0.3466	0.6810	0.3812	0.4428	0.3612	0.6982	0.3901	0.4519
	CAUM	0.3513	0.6819	0.3892	0.4557	0.3723	0.7059	0.4099	0.4698
	MINS	0.3562	0.6863	0.3907	0.4598	0.3707	0.7068	0.4102	0.4744
Temporal NRS	PP-Rec	0.3909	0.7112	0.4298	0.4966	0.4179	0.7302	0.4470	0.5098
	TCCM	0.4198	0.7235	0.4651	0.5291	0.4402	0.7499	0.4896	0.5467
Ours	CAST-Rec	**0.4513**	**0.7303**	**0.4992**	**0.5419**	**0.4697**	**0.7521**	**0.5186**	**0.5609**
	Improv.	7.50%	0.94%	7.33%	2.42%	6.70%	0.29%	5.92%	2.60%

Bold font is used to denote the best performance, while underlined font represents the second-best performance.

— **Area Under the Curve (AUC)** takes both the true-positive rate and false-positive rate into account. A model with a higher AUC is better at distinguishing between positively interacted items and negative items.
— **Normalized Discounted Cumulative Gain (NDCG)**@K accounts for the position of target items in the top-K ranked recommendation list and assigns higher importance to items in the front positions. Following previous works, we consider both NDCG@5 and NDCG@10, where a higher score represents better ranking performance.

4.1.4 Implementation Details. Our proposed CAST-Rec model is implemented using PyTorch and is trained on a single NVIDIA GeForce RTX 3090 GPU. The learning rate for CAST-Rec is selected from the values [5e-3, 3e-3, 1e-3, 5e-4, 3e-4, 1e-4], while the batch size is chosen from [64, 128, 256], and the dimensionality is selected from [8, 16, 32]. The model employs a dropout rate of 0.1, with a maximum sequence length of 5. We utilize two heads for the multi-head attention layers in CAST-Rec. For a fair comparison, all other baseline models are also trained on a single NVIDIA GeForce RTX 3090 GPU. Additionally, we adopt LLaMA-2-7B, LLaMA-2-13B, and Vicuna-13B models from Hugging Face for encoding time-invariant information. Specifically, the result as detailed in Table 2 is based on the inference on Vicuna-13B. The LLM inference is conducted using one NVIDIA A800 GPU. This setup ensures that the performance of CAST-Rec and the baseline models is evaluated under comparable hardware conditions, allowing for a rigorous assessment of the proposed model's capabilities.

4.2 Recommendation Performance (RQ1)

In this section, we evaluate the recommendation performance of CAST-Rec by comparing it with nine SOTA NRSs. These baseline models include a range of approaches, from traditional news content-matching models to sequential and temporal news recommendation models. We conducted experiments using widely recognized real-world news recommendation datasets, MIND-small and MIND-large, to ensure a robust evaluation. We compare the models based on three key metrics: MRR, AUC, and Top-K NDCG@K. Consistent with previous studies [4, 22, 44], we set K = [5, 10] for our evaluations. The comprehensive comparison aims to highlight how CAST-Rec performs relative to existing models in capturing user preferences and delivering relevant news recommendations.

Through experiments on MIND-small and MIND-large datasets, we provide insights into the effectiveness of our model across different dataset sizes.

We present the experimental results in Table 2. When compared to the best performance of previous models, specifically TCCM [4], our proposed CAST-Rec achieves notable improvements. Specifically, CAST-Rec outperforms TCCM by 7.50% and 6.70% in MRR and by 7.33% and 5.92% in NDCG@5 on the MIND-small and MIND-large datasets, respectively. These advancements can be attributed to two primary factors:

— *Modeling Evolving Causal Dependencies*: CAST-Rec effectively captures the evolving causal relationships between exposed news content and two time-variant covariates, thereby enhancing the modeling of user preferences over time. We achieve this by modeling both intra- and inter-dependencies among exposed news, news popularity, and news recency using masked self-attention and cross-attention layers within multi-input causal blocks.
— *Leveraging LLMs*: We utilize the semantic understanding capabilities of LLMs to generate concise summaries that encapsulate users' essential interests, which are then continuously refined over time. By predicting these distilled interests, the system is more likely to select the top-K news items that satisfy user preferences. The contribution and detailed analysis of using LLMs are discussed in Section 4.3.

Despite the significant improvements in MRR and NDCG@5, the enhancements in AUC and NDCG@10 are relatively modest, limited to 0.94% and 0.29% in AUC, 2.42% and 2.60% in NDCG@10 on the two datasets, respectively. This indicates a reduced improvement in the overall ranking of all candidate news items compared to the enhancement seen for the news items ranked at the top. We attribute this to the following two factors: First, CAST-Rec emphasizes items in the front positions due to a relatively small sequence length (i.e., 5 in experiments). Second, the baselines already demonstrate excellent overall ranking performance, making further improvements challenging. In addition, compared to MIND-small, the improvements on MIND-large are not as significant. A possible reason for this diminished improvement is that the MIND-large dataset contains more inactive users, resulting in a more long-tailed distribution, as evidenced in Table 1 and Figure 5.

Table 2 also provides insights into the impact of incorporating temporal information in news recommendations. Traditional NRSs (DKN [34], NPA [44], NRMS [45], NAML [43]) rely on personalized attention mechanisms to predict user preferences. However, they face performance bottlenecks. As the best model in the first category, NAML [43] achieves around 0.37 in the MIND-small dataset and 0.38 in the MIND-large dataset, with regard to NDCG@5. The second category of NRSs, including LSTUR [1], CAUM [23], and MINS [36], introduces sequential patterns in user interactions. These models capture the order of user actions but do not fully account for the time-variant factors related to the news items. For example, MINS [36] enhances recommendation performance by 5.62% in MIND-small and 7.30% in MIND-large datasets with regard to NDCG@5 by extracting multiple user interests using several GRUs from their browsing history. Temporal NRSs, such as PP-Rec [22] and TCCM [4], achieve the highest performance by modeling both the sequential and temporal aspects of evolving user preferences. These models consider not only the order of interactions but also the changing nature of user interests and news popularity over time. Specifically, PP-Rec [22] makes predictions by combining popularity scores with personalized matching scores, while TCCM [4] further advances this approach with causal interventions, leading to significant performance improvements. TCCM [4] achieves the best performance among baselines with an NDCG@5 score of 0.4651 in the MIND-small dataset, respectively. These insights underscore the importance of integrating temporal information into NRSs to better capture the evolving nature of user interest and news content.

4.3 Different LLMs (RQ2)

In this section, we delve into the performance of our proposed CAST-Rec framework when enhanced with different LLMs, conducting both ablation and case studies for analysis. Using metrics such as MRR, AUC, NDCG@5, and NDCG@10, we quantitatively assess recommendation performance under the enhancements of different LLMs. Additionally, we conduct a case study to closely examine the details in the summaries generated by these LLMs.

4.3.1 Ablation Study. We begin with an ablation study on several LLMs of different scales, demonstrating the effectiveness of introducing LLMs in our CAST-Rec. In particular, we consider four different models, i.e., LLaMA-2-7B [30], LLaMA-3-8B [20], LLaMA-2-13B [30], and Vicuna-13B [33].

- —LLaMA-2-7B [30] is an autoregressive model in the LLaMA series [29] that utilizes large-scale transformer architecture. LLaMA-2 [30] performs well in various natural language tasks through **Supervised Fine-tuning (SFT)** and **Reinforcement Learning with Human Feedback (RLHF)** to align generated texts with user instructions. LLaMA-2-7B consists of 7 billion parameters, balancing computational costs and semantics processing capability.
- —LLaMA-3-8B [20] is an autoregressive model trained with 8 billion parameters, building on its predecessors in the LLaMA family. It is also fine-tuned using SFT and RLHF, featuring an expanded context window and enhanced reasoning capabilities, making it particularly useful for complex natural language processing tasks.
- —Vicuna-13B [33] is fine-tuned on 13 billion parameters using 125,000 user-shared conversational data. This enables it to achieve impressive performance in various natural language tasks, such as question answering.
- —LLaMA-2-13B [30] presents superior performance in various natural language tasks compared to LLaMA-2-7B, owing to its larger parameter scale of 13 billion.

We replace the method for summarizing time-invariant user interests from browsing history with the aforementioned four LLMs, alongside the performance of random user interest initialization (without summarization, denoted as "random" in Table 3). The performance of CAST-Rec using these five different methods to capture time-invariant user interests is evaluated using MRR, AUC, NDCG@5, and NDCG@10, as shown in Table 3. Due to the significant computational costs, we limited our experiments to the MIND-small dataset. Our findings are as follows:

4.3.2 Case Study. To control the quality of the time-invariant covariates extracted by LLMs, in addition to using the hidden states, we also examine the corresponding generated texts. Specifically, we randomly select two users, `user 21` and `user 883`, from the MIND-small dataset and compare the texts generated by different LLMs.

In Figure 6, the red squares represent the news titles arranged in the order of the user's browsing history, while the blue squares include the texts generated by LLMs. Content within the same news topic is highlighted using the same text color, while incorrect topics generated by the LLMs are specifically highlighted in gray. `User 21` appears to prefer news related to social issues (red), celebrities (green), and events in the United States, especially in the Bay Area (blue). In contrast, `user 883` shows a broader range of interests, favoring news about politics (red), entertainment (green), sports (blue), and lifestyles (purple). It is observed that LLMs with smaller parameter sizes (LLaMA-2-7B and LLaMA-3-8B) are more prone to generating incorrect topics, leading to poorer performance. In contrast, larger LLMs can effectively capture users' time-invariant preferences through concise sentences, thereby improving user modeling in news recommendations. Moreover, LLMs are adept at summarizing various entities within a single topic across different news titles

Table 3. Ablation Studies Comparing the Recommendation Performance of Random Initialization versus Initialization Using Summaries Generated by Different LLMs for the Time-invariant User Interest within Our CAST-Rec Framework, Evaluated on the MIND-Small Dataset

Models	MRR	AUC	NDCG@5	NDCG@10
random	0.4012	0.7109	0.4315	0.4661
LLaMA-2-7B	0.4295	0.7149	0.4619	0.5007
	+7.05%	+0.56%	+7.05%	+7.42%
LLaMA-3-8B	0.4313	0.7183	0.4622	0.5019
	+7.50%	+1.04%	+7.11%	+7.68%
LLaMA-2-13B	0.4487	0.7201	0.4893	0.5272
	+11.84%	+1.29%	+13.40%	+13.11%
Vicuna-13B	0.4513	0.7303	0.4992	0.5419
	+12.49%	+2.73%	+15.69%	+16.26%

using general terms, which helps address the challenge of managing diverse textual representations in recommendation systems. Additionally, these LLM-generated summaries are more informative than the annotated categories of each news article.

4.4 Hyperparameter Study (RQ3)

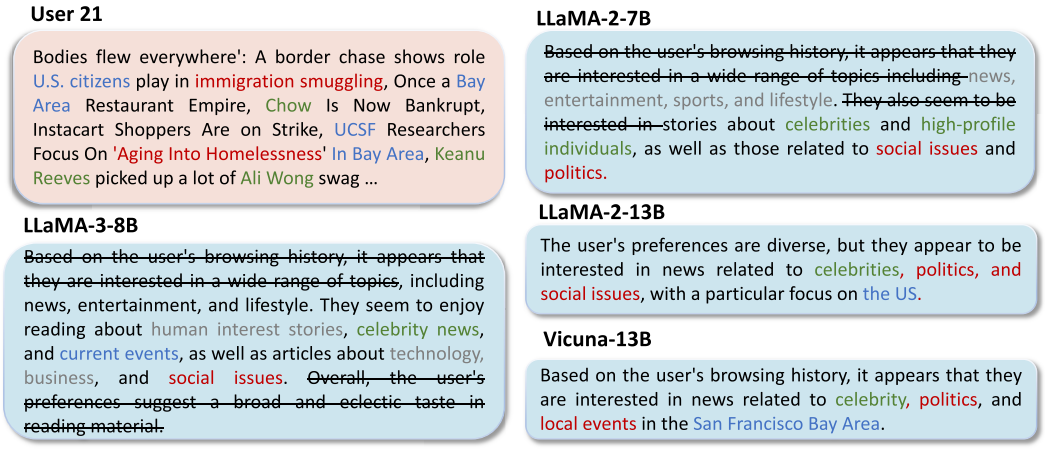
In this section, we have conducted studies on two hyperparameters to investigate their influence on recommendation performances: the dimension of hidden representations and the max length of user historical browsing sequences.

4.4.1 Different Dimensions Influence on Recommendation. The left plot in Figure 7 illustrates the effect of hidden representation dimensions on recommendation performance, measured by NDCG scores at the top-5 and top-10 lists (NDCG@5, NDCG@10). A dimension of 32 yields the best performance. This might be because a lower dimension (i.e., 8) lacks the capacity to capture the complex dependencies between variables, while a higher dimension (i.e., 128) suffers from data sparsity or difficulty in tuning other hyperparameters. The dimension is set to 32 in our experiments.

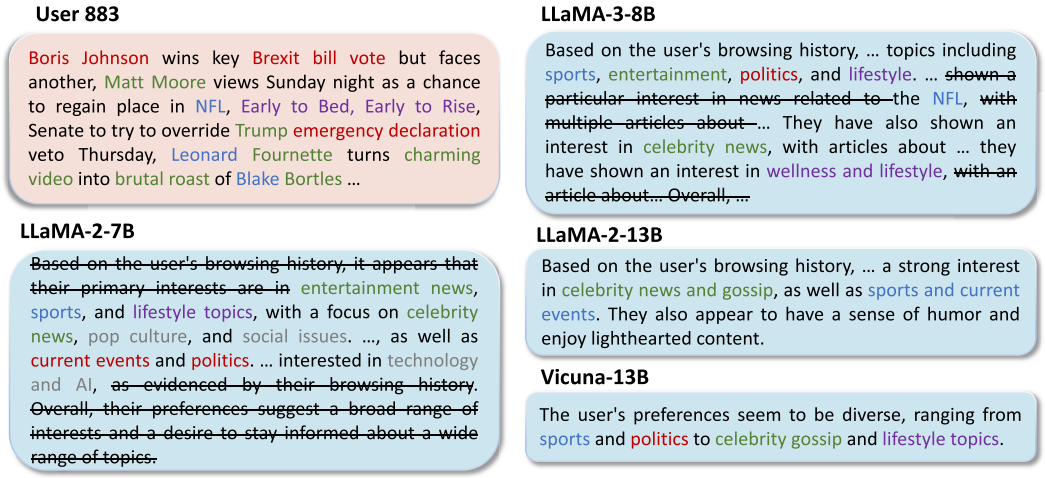
4.4.2 Different Sequence Lengths Influence on Recommendation. When modeling with multi-input causal blocks, we first divide the user browsing history into a series of subsequences. The right plot in Figure 7 shows the effects of the maximum length of these subsequences on recommendation performance, measured by NDCG@5 and NDCG@10. Short sequences fragment the sequential information in users' click behaviors, while longer sequences tend to be more expressive. Performance plateaus when the maximum length is set to 5 and peaks at a length of 15. However, due to time and memory limitations, we chose a maximum length of 5 in our experiments.

4.5 Efficiency Analysis (RQ4)

Time Complexity. The time complexity of CAST-Rec comes from the following processes: (1) time-aware modeling with L number of multi-input causal blocks (Section 3.3); (2) recommendations on $|\mathcal{V}|$ number of candidate news items (Section 3.4). Here, we use T to represent the length of user's browsing history, d to represent the hidden dimension in each respective model. For the first process, the time complexity in each causal block is mainly contributing to the multi-head self-/cross-attention layers, with $O(T^2Nd)$ each and $O(3T^2Nd)$ in total, where N denotes the number



(a) Case study of user 21 with four different LLMs.

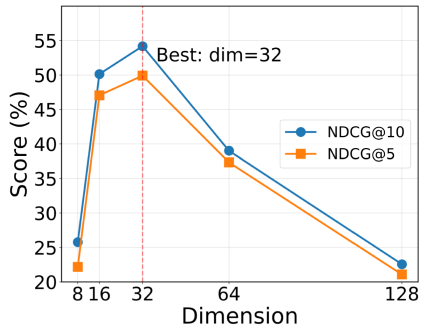


(b) Case study of user 883 with four different LLMs.

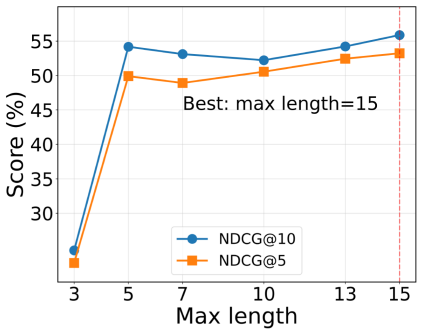
Fig. 6. Case studies of the time-invariant covariate summarized by different LLMs for users.

of heads. Subsequently, the feed-forward layer consumes $O(Td^2)$. The L numbers of causal blocks with three inputs inside each multiply these values, yielding $O(3L(3T^2Nd + Td^2))$. For the second process, the matrix factorization in user satisfaction score calculating costs $O(2|\mathcal{V}|d)$. In total, the time complexity for CAST-Rec is $O(2|\mathcal{V}|d + 3L(3T^2Nd + Td^2))$. Given the number of heads is set as 2 (i.e., $N = 2$) in our proposed CAST-Rec, and the number of multi-input causal blocks is set as 2 (i.e., $L = 2$), the time complexity can be simplified as $O(|\mathcal{V}|d + T^2d + Td^2)$.

Computational Efficiency. We provide the parameter sizes of different components in Table 4, showing that the primary computational and memory costs stem from the adoption of LLMs. In CAST-Rec, both the computational and time overhead can be mitigated by asynchronously utilizing APIs for LLMs. We conducted our experiments by deploying open source LLMs locally; however, our model is designed to accommodate API usage in scenarios with limited time and computational resources.



(a) Recommendation performance w.r.t different `Dimension` in MIND-small.



(b) Recommendation performance w.r.t. different `Max length` of browsing sequences in MIND-small.

Fig. 7. The influence of different values of hyperparameters `dimension` and the `maximum sequence length` on recommendation performances evaluated by NDCG@5 and NDCG@10.

Table 4. Parameter Sizes of the Different Components in CAST-Rec Using LLaMA-3-8B

Component	Parameter Sizes
LLM (depends on the model adopted)	8B
Causal transformer blocks (×3)	0.02 MB
Recommendation layer	2.39 MB
User and item embeddings	119.04 MB

5 Related Works

This section reviews previous related works on news recommendations in the lines of both traditional and LLM-based models. Additionally, we also survey the use of causality in recommendations.

5.1 News Recommendations

The task of news recommendation has received considerable attention, particularly with the current boom in LLMs that have transformed natural language processing. In the context of recommending news, the time-variant factors play a more crucial role compared to the other types of recommender systems, due to the need to consider the recency, popularity, and the steam of new content [25]. Traditional methods have laid the groundwork of news recommending. Recently, the emergence of LLMs has opened new avenues for enhancing the recommendation of news items by leveraging their advanced semantic understanding and generalization capability. This section delves into both the traditional recommenders and LLM-based approaches, respectively.

5.1.1 Traditional News Recommendations. Traditional methods [1, 44] address the challenge of news recommendation with neural sequential models and **Pre-trained Language Models (PLMs)** [17], such as BERT [7], GRU [5], and attention mechanism [32]. For example, NewsBERT [48] proposes a teacher–student framework to jointly learn from and distill complex user behavior patterns to make efficient news recommendations. Similarly, FeedRec [47] adeptly captures both positive and negative user preferences from explicit and implicit feedback using attention modules. Building on the groundwork established by these traditional news recommenders, researchers [12, 24, 46, 58] also delved into the new paradigm of news learning—prompt learning, which sets the stage for advancements in news recommendation powered by LLMs. For example, Li et al. [14]

approach news recommendation as a text-to-text language processing task and prompts the PLMs to personalized preferences.

5.1.2 LLM-based News Recommendations. Recently, the surge in popularity of LLMs has led to their extensive integration into NRSs [16, 17, 27, 59]. These studies utilize the sophisticated language capabilities of LLMs. For example, Liu et al. [16] and Zheng et al. [59] employ LLMs as a summarizer to condense news content and user behaviors, respectively. Additionally, the spotlight is on the effectiveness of using LLMs in news recommendations due to concerns about environmental sustainability [18, 19, 50]. However, our approach diverges from previous research by utilizing LLMs to facilitate the modeling of causal relationships in complex user behavior for user modeling in news recommendations. Furthermore, this article highlights the untapped potential of LLMs for news recommendations.

5.2 Causality in Recommendations

Causality has been demonstrated to be capable of improving the accuracy [10, 38, 52, 55, 56], explainability [15, 42, 51, 54], and fairness [13, 40, 41], and more of recommendation systems and machine learning [39, 51]. These methods can be divided into two distinct categories based on their underlying rationales. (i) The first category of methods enhances recommendations by identifying the cause–effect relationships rather than mere correlations among various elements within the recommendation process, such as users, items, and user features [37]. CaDSI [38] unravels user intents by explicitly learning the causal relationships by semantic-aware representation in recommendations. Zhang et al. [57] address popularity bias in recommendation system with causal intervention. (ii) Another line of approaches to enhance recommendation with causality is mitigating confounding effects in recommendations. Given that recommendation systems often contain covariates arising from noises and redundant correlations, their predictions can be distorted by the confounding effects that these covariates produce. Yang et al. [52] advance recommendations via utilizing counterfactual examples to deconfound the negative effects brought by intricate multi-typed user behaviors. In this article, we study the causality within the context of sequential news recommendation, where item popularity and user features introduce time-variant and time-invariant confounding effects, respectively. To the best of our knowledge, we are the first to propose a framework that simultaneously addresses both time-variant and time-invariant covariates in recommendation systems.

6 Conclusion

In this work, we approach news recommendations from a causal perspective and propose CAST-Rec to estimate the evolving causal influence of exposed news items, thereby enhancing next-item predictions. To carefully capture causal influences in the context of news recommendations, we model both time-invariant and time-variant covariates. We utilize LLMs to extract time-invariant covariates from user profiles and historical browsing contents. For time-variant covariates, we account for the influences of evolving popularity and recency on exposed news items using a series of transformer-based causal blocks. Our proposed CAST-Rec has been extensively tested on two real-world news datasets, consistently outperforming a broad range of existing news recommenders. These results demonstrate the effectiveness of integrating LLMs for capturing time-invariant information and highlight the importance of time-variant covariates.

References

[1] Mingxiao An, Fangzhao Wu, Chuhan Wu, Kun Zhang, Zheng Liu, and Xing Xie. 2019. Neural news recommendation with long- and short-term user representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 336–345. DOI: https://doi.org/10.18653/v1/P19-1033

[2] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal* 20, 6 (2017), 606–634.

[3] Wei Cai, Fuli Feng, Qifan Wang, Tian Yang, Zhenguang Liu, and Congfu Xu. 2023. A causal view for item-level effect of recommendation on user preference. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM '23)*. ACM, New York, NY, 240–248. DOI: https://doi.org/10.1145/3539597.3570461

[4] Yewang Chen, Weiyao Ye, Guipeng Xv, Chen Lin, and Xiaomin Zhu. 2023. TCCM: Time and content-aware causal model for unbiased news recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. ACM, New York, NY, 3778–3782. DOI: https://doi.org/10.1145/3583780.3615272

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555. Retrieved from https://arxiv.org/abs/1412.3555

[6] Gianmarco De Francisci Morales, Aristides Gionis, and Claudio Lucchese. 2012. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining (WSDM '12)*. ACM, New York, NY, 153–162. DOI: https://doi.org/10.1145/2124295.2124315

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805. Retrieved from https://arxiv.org/abs/1810.04805

[8] Florent Garcin, Christos Dimitrakakis, and Boi Faltings. 2013. Personalized news recommendation with context trees. In *Proceedings of the 7th ACM Conference on Recommender Systems*, 105–112.

[9] Linmei Hu, Siyong Xu, Chen Li, Cheng Yang, Chuan Shi, Nan Duan, Xing Xie, and Ming Zhou. 2020. Graph neural news recommendation with unsupervised preference disentanglement. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4255–4264.

[10] Sirui Huang, Qian Li, Xiangmeng Wang, Dianer Yu, Guandong Xu, and Qing Li. 2024. Counterfactual debasing for multi-behavior recommendations. In *Proceedings of the Database Systems for Advanced Applications: 29th International Conference (DASFAA '24)*, Part III. Springer-Verlag, Berlin, 164–179. DOI: https://doi.org/10.1007/978-981-97-5555-4_11

[11] Haoxuan Li, Kunhan Wu, Chunyuan Zheng, Yanghao Xiao, Hao Wang, Zhi Geng, Fuli Feng, Xiangnan He, and Peng Wu. 2024. Removing hidden confounding in recommendation: A unified multi-task learning approach. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '24)*. Curran Associates Inc., Red Hook, NY, Article 2380, 13 pages.

[12] Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. MINER: Multi-interest matching network for news recommendation. In *Findings of the Association for Computational Linguistics (ACL '22)*, 343–352.

[13] Qian Li, Xiangmeng Wang, Zhichao Wang, and Guandong Xu. 2023. Be causal: De-biasing social network confounding in recommendation. *ACM Transactions on Knowledge Discovery from Data* 17, 1 (Feb. 2023), Article 14, 1–23. DOI: https://doi.org/10.1145/3533725

[14] Xinyi Li, Yongfeng Zhang, and Edward C. Malthouse. 2023. Pbnr: Prompt-based news recommender system. arXiv:2304.07862. Retrieved from https://arxiv.org/abs/2304.07862

[15] Yicong Li, Hongxu Chen, Yile Li, Lin Li, Philip S. Yu, and Guandong Xu. 2023. Reinforcement learning based path exploration for sequential explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 11 (2023), 11801–11814. DOI: https://doi.org/10.1109/TKDE.2023.3237741

[16] Qijiong Liu, Nuo Chen, Tetsuya Sakai, and Xiao-Ming Wu. 2024. ONCE: Boosting content-based recommendation with both open- and closed-source large language models. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24)*. ACM, New York, NY, 452–461. DOI: https://doi.org/10.1145/3616855.3635845

[17] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiao-Ming Wu. 2022. Boosting deep CTR prediction with a plug-and-play pre-trainer for news recommendation. In *Proceedings of the 29th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2823–2833. Retrieved from https://aclanthology.org/2022.coling-1.249

[18] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiao-Ming Wu. 2023. Only encode once: Making content-based news recommender greener. arXiv:2308.14155. Retrieved from https://arxiv.org/abs/2308.14155

[19] Qijiong Liu, Jieming Zhu, Quanyu Dai, and Xiao-Ming Wu. 2024. Benchmarking news recommendation in the era of green AI. In *Companion Proceedings of the ACM on Web Conference 2024 (WWW '24)*. ACM, New York, NY, 971–974. DOI: https://doi.org/10.1145/3589335.3651472

[20] Meta. 2024. Build the future of AI with Meta Llama 3. Retrieved from https://llama.meta.com/llama3/

[21] Judea Pearl. 2009. Causal inference in statistics: An overview. *Statistics Surveys* 3 (2009), 96–146. DOI: https://doi.org/10.1214/09-SS057

[22] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. PP-Rec: News recommendation with personalized user interest and time-aware news popularity. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5457–5467.

[23] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2022. News recommendation with candidate-aware user modeling. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1917–1921.

[24] Tao Qi, Fangzhao Wu, Chuhan Wu, Peiru Yang, Yang Yu, Xing Xie, and Yongfeng Huang. 2021. HieRec: Hierarchical user interest modeling for personalized news recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, 5446–5456. DOI: https://doi.org/10.18653/v1/2021.acl-long.423

[25] Shaina Raza and Chen Ding. 2022. News recommender system: A review of recent progress, challenges, and opportunities. *Artificial Intelligence Review* 55, 1 (2022), 749–800.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. Retrieved from https://arxiv.org/abs/1205.2618

[27] Kaize Shi, Xueyao Sun, Dingxian Wang, Yinlin Fu, Guandong Xu, and Qing Li. 2025. LLaMA-E: Empowering E-commerce authoring with object-interleaved instruction following. In *Proceedings of the 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, Abu Dhabi, UAE, 870–885. Retrieved from https://aclanthology.org/2025.coling-main.58/

[28] Gabriele Sottocornola, Panagiotis Symeonidis, and Markus Zanker. 2018. Session-based news recommendations. In *Companion Proceedings of the Web Conference 2018*, 1395–1399.

[29] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. arXiv:2302.13971. Retrieved from https://arxiv.org/abs/2302.13971

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv:2307.09288. Retrieved from https://arxiv.org/abs/2307.09288

[31] Michele Trevisiol, Luca Maria Aiello, Rossano Schifanella, and Alejandro Jaimes. 2014. Cold-start news recommendation with domain-dependent browse graph. In *Proceedings of the 8th ACM Conference on Recommender Systems*, 81–88.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).

[33] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality. Retrieved from https://lmsys.org/blog/2023-03-30-vicuna/

[34] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, 1835–1844.

[35] Jingkun Wang, Yipu Chen, Zichun Wang, and Wen Zhao. 2021. Popularity-enhanced news recommendation with multi-view interest representation. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21)*. ACM, New York, NY, 1949–1958. DOI: https://doi.org/10.1145/3459637.3482462

[36] Rongyao Wang, Shoujin Wang, Wenpeng Lu, and Xueping Peng. 2022. News recommendation via multi-interest news sequence modelling. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7942–7946.

[37] Wenjie Wang, Yang Zhang, Haoxuan Li, Peng Wu, Fuli Feng, and Xiangnan He. 2023. Causal recommendation: Progresses and future directions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23)*. ACM, New York, NY, 3432–3435. DOI: https://doi.org/10.1145/3539618.3594245

[38] Xiangmeng Wang, Qian Li, Dianer Yu, Peng Cui, Zhichao Wang, and Guandong Xu. 2023. Causal disentanglement for semantic-aware intent learning in recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 10 (2023), 9836–9849. DOI: https://doi.org/10.1109/TKDE.2022.3159802

[39] Xiangmeng Wang, Qian Li, Dianer Yu, Wei Huang, Qing Li, and Guandong Xu. 2024. Neural causal graph collaborative filtering. *Information Sciences* 677 (Aug 2024), 120872. DOI: https://doi.org/10.1016/j.ins.2024.120872

[40] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2023. Constrained off-policy learning over heterogeneous information for fairness-aware recommendation. *ACM Transactions on Recommender Systems* 2, 4 (2023), 1–27. DOI: https://doi.org/10.1145/3629172

[41] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2024. Counterfactual explanation for fairness in recommendation. *ACM Transactions on Information Systems* 42, 4 (Mar. 2024), Article 106, 1–30. DOI: https://doi.org/10.1145/3643670

[42] Xiangmeng Wang, Qian Li, Dianer Yu, Qing Li, and Guandong Xu. 2024. Reinforced path reasoning for counterfactual explainable recommendation. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3443–3459. DOI: https://doi.org/10.1109/TKDE.2024.3354077

[43] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with attentive multi-view learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 3863–3869.

[44] Chuhan Wu, Fangzhao Wu, Mingxiao An, Jianqiang Huang, Yongfeng Huang, and Xing Xie. 2019. NPA: Neural news recommendation with personalized attention. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2576–2584.

[45] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie.2019. Neural news recommendation with multi-head self-attention. 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)*, 6389–6394.

[46] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1652–1656.

[47] Chuhan Wu, Fangzhao Wu, Tao Qi, Qi Liu, Xuan Tian, Jie Li, Wei He, Yongfeng Huang, and Xing Xie. 2022. FeedRec: News feed recommendation with various user feedbacks. In *Proceedings of the ACM Web Conference 2022 (WWW '22)*. ACM, New York, NY, 2088–2097. DOI: https://doi.org/10.1145/3485447.3512082

[48] Chuhan Wu, Fangzhao Wu, Yang Yu, Tao Qi, Yongfeng Huang, and Qi Liu. 2021. NewsBERT: Distilling pre-trained language model for intelligent news application. In *Findings of the Association for Computational Linguistics (EMNLP '21)*. Association for Computational Linguistics, Punta Cana, Dominican Republic, 3285–3295. DOI: https://doi.org/10.18653/v1/2021.findings-emnlp.280

[49] 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*. Association for Computational Linguistics, 3597–3606. DOI: https://doi.org/10.18653/v1/2020.acl-main.331

[50] Jiahao Wu, Qijiong Liu, Hengchang Hu, Wenqi Fan, Shengcai Liu, Qing Li, Xiao-Ming Wu, and Ke Tang. 2023. Leveraging large language models (LLMs) to empower training-free dataset condensation for content-based recommendation. arXiv:2310.09874. Retrieved from https://arxiv.org/abs/2310.09874

[51] Guandong Xu, Tri Dung Duong, Qian Li, Shaowu Liu, and Xianzhi Wang. 2020. Causality learning: A new perspective for interpretable machine learning. arXiv:2006.16789. Retrieved from https://arxiv.org/abs/2006.16789

[52] Haoran Yang, Hongxu Chen, Sixiao Zhang, Xiangguo Sun, Qian Li, Xiangyu Zhao, and Guandong Xu. 2023. Generating counterfactual hard negative samples for graph contrastive learning. In *Proceedings of the ACM Web Conference 2023 (WWW '23)*. ACM, New York, NY, 621–629. DOI: https://doi.org/10.1145/3543507.3583499

[53] Yu Yang, Hongzhi Yin, Jiannong Cao, Tong Chen, Quoc Viet Hung Nguyen, Xiaofang Zhou, and Lei Chen. 2023. Time-aware dynamic graph embedding for asynchronous structural evolution. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2023), 9656–9670.

[54] Dianer Yu, Qian Li, Xiangmeng Wang, Qing Li, and Guandong Xu. 2024. Counterfactual explainable conversational recommendation. *IEEE Transactions on Knowledge and Data Engineering* 36, 6 (2024), 2388–2400. DOI: https://doi.org/10.1109/TKDE.2023.3322403

[55] Dianer Yu, Qian Li, Xiangmeng Wang, and Guandong Xu. 2023. Deconfounded recommendation via causal intervention. *Neurocomputing* 529 (2023), 128–139.

[56] Dianer Yu, Qian Li, Hongzhi Yin, and Guandong Xu. 2023. Causality-guided graph learning for session-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23)*. ACM, New York, NY, 3083–3093. DOI: https://doi.org/10.1145/3583780.3614803

[57] 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 (SIGIR '21)*. ACM, New York, NY, 11–20. DOI: https://doi.org/10.1145/3404835.3462875

[58] Zizhuo Zhang and Bang Wang. 2023. Prompt learning for news recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 227–237.

[59] Zhi Zheng, WenShuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. 2024. Harnessing large language models for text-rich sequential recommendation. In *Proceedings of the ACM on Web Conference 2024 (WWW '24)*. ACM, New York, NY, 3207–3216. DOI: https://doi.org/10.1145/3589334.3645358

Received 15 September 2024; revised 20 December 2024; accepted 20 February 2025