

Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue

Wenjie Wang
wenjiewang96@gmail.com
National University of Singapore

Fuli Feng*
fulifeng93@gmail.com
Sea-NExT Joint Lab, Singapore
National University of Singapore

Xiangnan He
hexn@ustc.edu.cn
University of Science and Technology
of China

Hanwang Zhang
hanwangzhang@ntu.edu.sg
Nanyang Technological University

Tat-Seng Chua
dcscts@nus.edu.sg
National University of Singapore

ABSTRACT

Recommendation is a prevalent and critical service in information systems. To provide personalized suggestions to users, industry players embrace machine learning, more specifically, building predictive models based on the click behavior data. This is known as the Click-Through Rate (CTR) prediction, which has become the gold standard for building personalized recommendation service. However, we argue that there is a significant gap between clicks and user satisfaction — it is common that a user is “cheated” to click an item by the attractive title/cover of the item. This will severely hurt user’s trust on the system if the user finds the actual content of the clicked item disappointing. What’s even worse, optimizing CTR models on such flawed data will result in the Matthew Effect, making the seemingly attractive but actually low-quality items be more frequently recommended.

In this paper, we formulate the recommendation models as a causal graph that reflects the cause-effect factors in recommendation, and address the clickbait issue by performing counterfactual inference on the causal graph. We imagine a counterfactual world where each item has only exposure features (*i.e.*, the features that the user can see before making a click decision). By estimating the click likelihood of a user in the counterfactual world, we are able to reduce the direct effect of exposure features and eliminate the clickbait issue. Experiments on real-world datasets demonstrate that our method significantly improves the post-click satisfaction of CTR models.

CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Learning from implicit feedback.

*Corresponding author: Fuli Feng (fulifeng93@gmail.com).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '21, July 11–15, 2021, Virtual Event, Canada

© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8037-9/21/07...\$15.00
<https://doi.org/10.1145/3404835.3462962>

KEYWORDS

Counterfactual, Clickbait Issue, Counterfactual Recommendation

ACM Reference Format:

Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, July 11–15, 2021, Virtual Event, Canada. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3404835.3462962>

1 INTRODUCTION

Recommender systems have been increasingly used to alleviate information overloading for users in a wide spectrum of information systems such as e-commerce [59], digital streaming [50], and social networks [19]. To date, the most recognized way for training recommender model is to optimize the Click-Through Rate (CTR), which aims to maximize the likelihood that a user clicks the recommended items. Despite the wide deployment of CTR optimization in recommender systems, we argue that the user experience may be hurt unintentionally due to the clickbait issue. That is, some items with attractive exposure features (*e.g.*, title and cover image) are easy to attract user clicks [20, 57], and thus are more likely to be recommended, but their actual content does not match the exposure features and disappoints the users. Such clickbait issue is very common, especially in the present era of self-media, posing great obstacles for the platform to provide high-quality recommendations (*cf.* Figure 4 for the evidence).

To illustrate, Figure 1 shows an example that a user clicks two recommended videos with observation of their exposure features only. After watching the video, *i.e.*, examining the video content after clicking, the user gives the ratings of whether like or dislike the recommendations. *Item2* receives a dislike since the title deliberately misleads the user to click it, whereas *item1* receives a like since its actual content matches the title and cover image, and satisfies the user. This reflects the possible (in fact, significant) gap between clicks and satisfaction — many clicks would end up with dissatisfaction since the click depends largely on whether the user is interested in the exposure features of the item.

Assuming that we can extract good content features that are indicative of item quality and even consistent with user satisfaction, can we address the discrepancy issue? Unfortunately, the answer is no. The reason roots in the optimization objective — CTR: when we train a recommender model to maximize the click likelihood of the

items with the clickbait issue, the model will learn to emphasize the exposure features and ignore the signal from other features, because the attractive exposure features are the causal reason of user clicks. This will aggravate the negative effect of clickbait issue, making these seemingly attractive but low-quality items be recommended more and more frequently.

To address the issue, a straightforward solution is to leverage the post-click feedback from users [32, 51], such as the like/dislike ratings and numeric reviews. However, the amount of such explicit feedback is much smaller than that of click data, since many users are reluctant to leave any feedback after clicks. In most real-world datasets, users have very few post-click feedback, making it difficult to utilize them to supplement the large-scale implicit feedback well. Towards a wider range of applications and broader impact, we believe that it is critical to solve the clickbait issue in recommender system based on the click feedback only, which is highly challenging and has never been studied before.

In this work, we approach the problem from a novel perspective of causal inference: if we can distinguish the effects of exposure features (pre-click) and content features (post-click) on the prediction, then we can reduce the effect of exposure features that cause the clickbait issue. Towards this end, we first build a causal graph that reflects the cause-effect factors in recommendation scoring (Figure 3(b)). Next, we estimate the *direct effect* of exposure features on the prediction score in a counterfactual world (Figure 3(c)), which imagines *what the prediction score would be if the item had only the exposure features*. During inference, we remove this direct effect from the prediction in the factual world, which presents the total effect of all item features. In the example of Figure 1, although *item1* and *item2* obtain similar scores in the factual world, the final score of *item2* will be largely suppressed, because its content features are disappointing and it is the deceptive exposure features that increase the prediction score in the factual world. We instantiate the framework on MMGCN [50], a representative multi-modal recommender model that can handle both exposure and content features. Extensive experiments on two widely used benchmarks show the superiority of the proposed framework, which significantly reduces the clickbait issue by only using the click feedback and recommends more satisfying items.

To sum up, the contributions of this work are threefold:

- We highlight the importance of mitigating the clickbait issue by using click data only and leverage a new causal graph to formulate the recommendation process.
- We introduce counterfactual inference into recommendation to mitigate the clickbait issue, and propose a counterfactual recommendation framework which can be applied to any recommender models with item features as inputs.
- We implement the proposed framework on MMGCN and conduct extensive experiments on two widely used benchmarks, which validate the effectiveness of our proposal.

2 TASK FORMULATION

In this section, we formulate the recommender training and the clickbait issue, followed by the task evaluation.

Recommender training. The target of recommender training is to learn a scoring function s_θ that predicts the preference of a

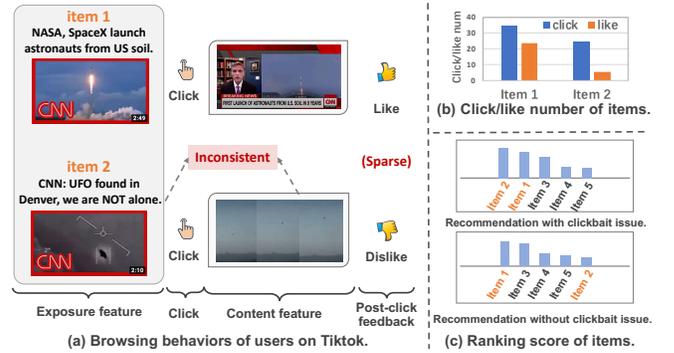


Figure 1: (a) Illustration of inconsistency between clicks and likes. (b) Number of clicks/likes on the two items where few clicks on item 2 end with likes. (c) Two recommendation lists with and without the clickbait issue, respectively.

user over an item. Formally, $Y_{u,i} = s_\theta(u, i)$ where u and i denote user features and item features, respectively. Specifically, item features $i = (e, t)$ include both exposure features e and content features t which are observed by users before and after clicks, respectively. θ denotes the model parameters which are typically learned from historical click data $\bar{\mathcal{D}} = \{(u, i, \bar{Y}_{u,i}) | u \in \mathcal{U}, i \in \mathcal{I}\}$, where $\bar{Y}_{u,i} \in \{0, 1\}$ denotes whether u clicks i ($\bar{Y}_{u,i} = 1$) or not ($\bar{Y}_{u,i} = 0$). \mathcal{U} and \mathcal{I} refer to the user set and item set, respectively. In this work, we use click to represent any type of implicit interactions for brevity, including purchase, watch, and download. Formally, the recommender training is:

$$\bar{\theta} = \arg \min_{\theta} \sum_{(u, i, \bar{Y}_{u,i}) \in \bar{\mathcal{D}}} l(s_\theta(u, i), \bar{Y}_{u,i}), \quad (1)$$

where $l(\cdot)$ denotes the recommendation loss such as cross-entropy loss [15]. During inference, the trained recommender model serves each user by ranking all items according to $Y_{u,i} = s_{\bar{\theta}}(u, i)$ and recommending the top-ranked ones to the user.

Clickbait Issue. The clickbait issue is recommending items with attractive exposure features but disappointing content features frequently. Formally, given item i with attractive exposure features but dissatisfying content, and item j with less attractive exposure features and satisfying content, the clickbait issue happens if:

$$s_{\bar{\theta}}(u, i = (e_i, t_i)) > s_{\bar{\theta}}(u, j = (e_j, t_j)), \quad (2)$$

where item i ranks higher than item j . That is, items with more attractive exposure features (e.g., *item2* in Figure 1) occupy the recommendation opportunities of items with satisfying content features (e.g., *item1* in Figure 1).

Consequently, the recommender models will recommend many items like i , which will hurt user experience and lead to more clicks that end with dislikes. And worse still, it forms a vicious spiral: in turn, such clicks aggravate the issue in future recommender training. In this work, we aim to break the vicious spiral by mitigating the clickbait issue during inference, i.e., forcing $Y_{u,i} < Y_{u,j}$ for more user satisfaction rather than a higher CTR. Furthermore, we solve the problem based on click feedback only, i.e., no post-click feedback is accessible during the recommender training.

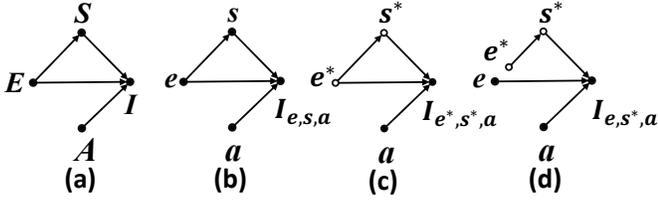


Figure 2: (a) An example of a causal graph where the individual income (I) is directly affected by the education (E), age (A), and skill (S); and skill is influenced by education. (b) A causal graph with particular realizations. (c) A causal intervention $do(E = e^*)$, where e^* denotes the reference value of E , e.g., no qualifications. (d) One counterfactual where S is set as s^* while keeping $E = e$ on the edge $E \rightarrow I$.

Evaluation. Distinct from the conventional recommender evaluation that treats all clicks in the testing period as positive samples [17, 50], we evaluate recommendation performance only over clicks that end with positive post-click feedback (i.e., likes) [52]. We do not use the clicks that lack post-click feedback due to the unawareness of user satisfaction. In addition, we believe that the recommendation performance on the selected clicks is able to validate the effectiveness of solving the clickbait issue. This is because a recommender model affected by the clickbait issue will fail on a portion of the selected clicks because they prefer to recommend items with more attractive exposure features but dissatisfying content features.

3 PRELIMINARY

We briefly introduce the concepts of counterfactual inference [37, 38] used in this paper, and refer readers to learn from the related works [35, 37, 43–45] for a comprehensive understanding.

Causal Graph. Causal graph describes the causal relations between variables by a directed acyclic graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where \mathcal{N} is the set of variables (i.e., nodes) and \mathcal{E} records the causal relations (i.e., edges). In the causal graph, capital letters and lowercase letters denote random variables (e.g., X) and the specific realizations of random variables (e.g., x), respectively. Figure 2(a) illustrates an example of a causal graph that represents the causal relations to the individual income: 1) the individual income (I) is directly affected by the education (E), age (A), and skill (S); and 2) indirectly affected by the education through a mediator S . According to the graph structure, a set of structural equations \mathcal{F} [38] can be used to measure how the variables are affected by their parents. For example, we can estimate the values of S and I from their parents by $\mathcal{F} = \{f_S(\cdot), f_I(\cdot)\}$. Formally,

$$\begin{cases} S_e = s = f_S(E = e), \\ I_{e,s,a} = f_I(E = e, S = s, A = a), \end{cases} \quad (3)$$

where $I_{e,s,a}$ denotes the income of one person who satisfies $E = e$, $S = s$, and $A = a$. $f_S(\cdot)$ and $f_I(\cdot)$ correspond to the structural equations of variable S and I , respectively, which can be learned from a set of observations [38].

Counterfactuals. Counterfactual inference [39] is a technique to estimate what the descendant variables would be if the value of one treatment variable were different with its real value in the

factual world. As shown in Figure 2(d), counterfactual inference can estimate *what the income of Joe would be if he only had the skill of a person without qualifications*. That is imagining a situation: I receives $E = e$ through $E \rightarrow I$, while S receives $E = e^*$ through $E \rightarrow S$ and other variables are fixed. Specifically, e can represent a bachelor degree while e^* denotes no qualifications. The key to counterfactual inference lies in performing external intervention [38] to control the value of S , which is termed as *do-operator*. Formally, $do(S = s^*)$ forcibly substitute s with $s^* = f_S(E = e^*)$ in the structural equation f_I , obtaining $I_{e,s^*,a} = f_I(E = e, S = s^*, A = a)$. Note that $do(S = s^*)$ does not affect the ascendant variables of S , i.e., E retains its real value e on the direct path $E \rightarrow I$.

Causal Effect. Causal effect¹ of one event with the treatment variable (e.g., $E = e$, obtaining a bachelor degree) on the response variable (e.g., I) measures the change of the response variable when the treatment variable changes from its reference value (e.g., e^*) to the expected value (e.g., e), which is also termed as *total effect* (TE). Formally, the TE of $E = e$ on I under situation $A = a$ is defined as:

$$\begin{aligned} \text{TE} &= I_{e,s,a} - I_{e^*,s^*,a}, \\ &= f_I(E = e, S = s, A = a) - f_I(E = e^*, S = s^*, A = a), \end{aligned} \quad (4)$$

where $I_{e^*,s^*,a}$ denotes the reference status of I when $E = e^*$, i.e., the outcome of the intervention $do(E = e^*)$ (see Figure 2(c)). Specifically, by viewing e^* as no qualifications, $I_{e^*,s^*,a}$ denotes the income of Joe if he hadn't got qualifications (i.e., $E = e^*$) at the age of a . Furthermore, the event affects the response variable through both the direct path between the two variables (e.g., $E \rightarrow I$) and the indirect path via mediators (e.g., $E \rightarrow S \rightarrow I$). A widely used decomposition of TE is $\text{TE} = \text{NDE} + \text{TIE}$, where NDE and TIE denote the *natural direct effect* and *total indirect effect* [37, 45], respectively.

In particular, NDE is the change of the response variable when only changing the treatment variable on the direct path, i.e., the mediators retain unchanged and still receive the reference value. For instance, the NDE of $E = e$ on I under situation a is the change of the income I when changing E from e^* to e and forcing $S = s^*$. Formally, the calculation of NDE relies on $do(S = s^*)$, which is:

$$\text{NDE} = I_{e,s^*,a} - I_{e^*,s^*,a}, \quad (5)$$

where $I_{e,s^*,a}$ is the income in a counterfactual world (see Figure 2(d)). Accordingly, the TIE of $E = e$ on I under situation $A = a$ can be obtained by subtracting NDE from TE [45]:

$$\text{TIE} = \text{TE} - \text{NDE} = I_{e,s,a} - I_{e,s^*,a}. \quad (6)$$

Generally, TIE is the change of the response variable when the mediators are changed from their reference values (e.g., $s^* = f_S(E = e^*)$) to the ones receiving the expected value (e.g., $s = f_S(E = e)$), and the value of the treatment variable on the direct path remains fixed (e.g., $E = e$ on $E \rightarrow I$).

4 COUNTERFACTUAL RECOMMENDATION

In this section, we introduce the causal graph of recommender systems, followed by the elaboration of counterfactual inference to mitigate the clickbait issue and the design of proposed counterfactual recommendation (CR) framework.

¹In this work, causal effect is defined at the unit level [37, 38], i.e., the effect is on one individual rather than a population.

4.1 Causal Graph of Recommender Systems

In Figure 3(a), we abstract the causal graph of existing recommender models where Y , U , I , E , and T denote the prediction score, user features, item features, exposure features, and content features, respectively. Accordingly, the existing recommender model (i.e., $s_\theta(\cdot)$) is abstracted as two structural equations $f_Y(\cdot)$ and $f_I(\cdot)$, which are formulated as:

$$Y_{u,i} = f_Y(U = u, I = i), \text{ where } i = I_{e,t} = f_I(E = e, T = t). \quad (7)$$

The two structural equations $f_Y(\cdot)$ and $f_I(\cdot)$ correspond to the main modules of the existing models, the scoring function (e.g., inner product function) and feature aggregation function (e.g., multi-layer perceptron (MLP) [15]), respectively. In particular, $f_I(\cdot)$ aims to extract the representative item features from its exposure and content features, which are then fed into $f_Y(\cdot)$ for making the prediction. The parameters of the equations (i.e., θ) are learned by minimizing the recommendation loss over historical data, so as to maximize the likelihood of the clicked items (i.e., Equation 1).

However, the causal graph of existing recommender models mismatches the generation process of the training data. In the user browsing process, users might click the items only because they are attracted by the exposure features². From the cause-effect view, there is a direct effect from the exposure features to the click behavior. As a result of ignoring such direct effect in the model, the feature aggregation function will inevitably emphasize the exposure features while ignoring the content features (see empirical results in Figure 8), in order to achieve a small loss on the clicked items with the clickbait issue.

To bridge this gap, we build a new causal graph by adding a direct edge from exposure features E to the prediction Y (Figure 3(b)). According to the new causal graph, the recommender model should capture the causal effect of exposure features on prediction Y through both the direct path ($E \rightarrow Y$) and the indirect path ($E \rightarrow I \rightarrow Y$). Formally, the abstract format of the model should be:

$$Y_{u,i,e} = f_Y(U = u, I = i, E = e), \text{ where } i = f_I(E = e, T = t). \quad (8)$$

In other words, when we design a recommender model that will be optimized over historical clicks through the CTR objective, its scoring function should directly take exposure features as one additional input.

4.2 Mitigating Clickbait Issue

While the new causal graph provides a more precise description of the cause-effect factors for recommendation scoring, the recommender model based on the new causal graph still suffers from the clickbait issue (in Equation 2). This is because the outcome of the response variable, i.e., $Y_{u,i,e}$, still accounts for the direct effect of exposure features. Consequently, the item (e.g., *item2* in Figure 1) with more attractive exposure features is still scored higher than the one with more satisfying content but less attractive exposure features. To mitigate the clickbait issue, we perform *CR inference* to reduce the direct effect of exposure features from the prediction $Y_{u,i,e}$, which is formulated as $Y_{u,i,e} - \text{NDE}$.

²Note that the click behavior can also be affected by other item features (i.e., I), e.g., the category and uploader of videos.

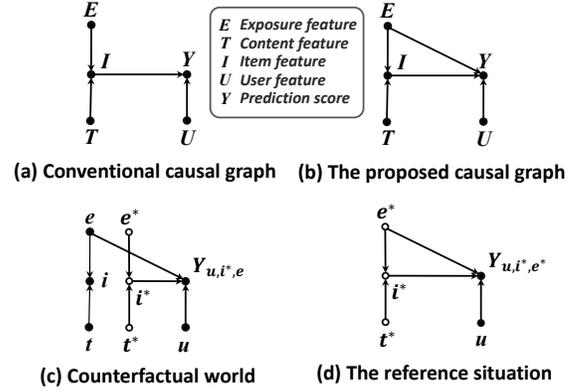


Figure 3: The causal graphs for conventional and counterfactual recommendations. * denotes the reference values.

Towards this end, we need to estimate the NDE of event $E = e$ on the response variable Y . In particular, we estimate the NDE under situation $U = u$ and $T = t^*$. As detailed in Section 3, the NDE is formulated as:

$$\begin{aligned} \text{NDE} &= Y_{u,t^*,e} - Y_{u,i^*,e^*} \\ &= f_Y(U = u, I = i^*, E = e) - f_Y(U = u, I = i^*, E = e^*), \end{aligned}$$

where $i^* = f_I(E = e^*, T = t^*)$, and e^* and t^* are the reference values of E and T , respectively. $f_Y(U = u, I = i^*, E = e)$ denotes the outcome of a counterfactual (see Figure 3(c)) where the treatment variable E is changed from e^* to e on the direct path (i.e., $E \rightarrow Y$) while remains its reference value on the indirect path (i.e., $E \rightarrow I \rightarrow Y$). That is, it estimates *what the prediction score would be if the item had only the exposure features* in a counterfactual world, i.e., to what extent the user is purely attracted by exposure features. In this task, the reference values e^* and t^* are treated as the status that the features are not given. Given the user features u , the second term Y_{u,i^*,e^*} (Figure 3(d)) is thus a constant for any items, i.e., Y_{u,i^*,e^*} will not affect the ranking of items for a user. Therefore, by subtracting the NDE of exposure features from $Y_{u,i,e}$, the prediction score of CR inference becomes:

$$Y_{CR} = Y_{u,i,e} - Y_{u,i^*,e^*}. \quad (9)$$

Intuitively, Y_{CR} reduces the NDE of exposure features and relies on the effect of the combined item features I for inference. The prediction score of the item with attractive exposure features but boring content (e.g., *item2* in Figure 1) will be largely suppressed during CR inference, because its only attractiveness is in the exposure features and the content features are dissatisfying. It will have a high prediction score in the counterfactual world (i.e., $Y_{u,i^*,e}$). Accordingly, the item with less attractive exposure features but satisfying content features (e.g., *item1* in Figure 1) will have a higher chance to be recommended because the satisfactory item features I will increase the prediction score in CR inference, which forces $s_\theta(u, i) < s_\theta(u, j)$ in Equation 2.

From the cause-effect view, CR inference subtracts the NDE of $E = e$ from the TE of $E = e$ and $T = t$. As introduced in Section 3, the TE of $E = e$ and $T = t$ on Y under situation $U = u$ can be calculated by $Y_{u,i,e} - Y_{u,i^*,e^*}$ where Y_{u,i^*,e^*} is the reference situation. Obviously, the prediction score of CR inference can be formulated as $Y_{CR} = \text{TE} - \text{NDE}$.

Note that we can estimate the NDE of $E = e$ on Y under the situation of $T = t^*$ or $T = t$ [37]. Changes of the situation can lead to minor difference in the estimation since the recommender models are typically non-linear [38, 45]. We select the situation of $T = t^*$ to avoid the leakage of exposure features. This is because, in the recommendation scenarios, the content features t might include some information in the exposure features e . For instance, the cover image may be a frame in the video, which might cause the leakage of e through the mediator I . Empirical evidence in Table 3 justifies the advantage of this choice.

4.3 CR Framework Design

Recall that the key to counterfactual inference lies in the learned structural equations. To enable CR inference, we thus need to design a recommender model according to the proposed causal graph in Figure 3(b) and an algorithm to learn the model parameters.

4.3.1 Model Design. According to Equation 8, the recommender model should consist of two functions: the scoring function $f_Y(U = u, I = i, E = e)$ and the feature aggregation function $f_I(E = e, T = t)$. As to the feature aggregation function, we can simply employ the one in existing models to encode the causal relations from E and T to I . We focus on upgrading the conventional scoring function $f_Y(U = u, I = i)$ to $f_Y(U = u, I = i, E = e)$.

- **Scoring Function.** A straightforward idea is to embed the additional input e into the conventional scoring function. However, this solution loses generality due to requiring careful adjustments for different recommender models. According to the universal approximation theorem [10], we could also implement $f_Y(\cdot)$ by a MLP with u , i , and e as the inputs. Nevertheless, it is hard to tune a MLP to achieve the comparable performance with the models wisely designed for the recommendation task [19, 40, 50].

Aiming to keep generality and leverage the advantages of existing models, the scoring function is implemented in a late-fusion manner [4, 35]:

$$f_Y(U = u, I = i, E = e) = f(Y_{u,i}, Y_{u,e}),$$

where $Y_{u,i} = f_Y(U = u, I = i)$ and $Y_{u,e} = f_Y(U = u, E = e)$ are the predictions from two conventional models with different inputs; and $f(\cdot)$ is a fusion function. $Y_{u,i}$ and $Y_{u,e}$ can be instantiated by any recommender models with user and item features as the inputs such as MMGCN [50] and VBPR [17]. In this way, we can simply adapt an existing recommender model to fit in the proposed causal graph by additionally implementing a fusion strategy, which can be easily achieved.

- **Fusion strategy.** Inspired by the prior studies [4, 35], we adopt one classic fusion strategy: Multiplication (MUL), formulated as:

$$Y_{u,i,e} = f_Y(U = u, I = i, E = e) = f(Y_{u,i}, Y_{u,e}) = Y_{u,i} * \sigma(Y_{u,e}),$$

where σ denotes the sigmoid function. It provides non-linearity for sufficient representation capacity of the fusion strategy, which is essential (see results in Table 5). Note that the proposed CR is general to any differentiable arithmetic binary operations and we compare more strategies in Table 5.

4.3.2 Model Training. Recall that the CR inference requires two predictions: $Y_{u,i,e}$ and $Y_{u,i^*,e}$. The target of model training is thus

twofold – learning parameters of the structural equations (*i.e.*, $f_Y(\cdot)$ and $f_I(\cdot)$) that can accurately estimate both $Y_{u,i,e}$ and $Y_{u,i^*,e}$. As such, we optimize a multi-task training objective over historical clicks to learn the model parameters, which is formulated as:

$$\mathcal{L} = \sum_{(u,i,\bar{Y}_{u,i}) \in \mathcal{D}} l(Y_{u,i,e}, \bar{Y}_{u,i}) + \alpha * l(Y_{u,e}, \bar{Y}_{u,i}), \quad (10)$$

where $\bar{Y}_{u,i}$ is the label for u and i , and α is a hyperparameter to tune the relative weight of two tasks. Recall that i^* indicates the recommender model doesn't take i as the input, and thus $Y_{u,e}$ can be seen as the learned prediction $Y_{u,i^*,e}$ based on the user features u and exposure features e in the counterfactual world.

- **CR Inference.** CR inference needs to calculate the predictions $Y_{u,i,e} = f(Y_{u,i}, Y_{u,e})$ and $Y_{u,i^*,e} = f(c_u, Y_{u,e})$ where c_u refers to the expectation constants of $Y_{u,I}$:

$$c_u = E(Y_{u,I}) = \frac{1}{|I|} \sum_{i \in I} Y_{u,i}, \quad (11)$$

which indicates that for each user, all the items share the same score c_u . Since the features of I are not given in $Y_{u,i^*,e}$, the model used to predict $Y_{u,i}$ ranks items with the same score c_u for user u . In this way, the results of CR inference will be calculated by:

$$Y_{CR} = Y_{u,i,e} - Y_{u,i^*,e} = Y_{u,i,e} - f(c_u, Y_{u,e}) = Y_{u,i,e} - c_u * \sigma(Y_{u,e}).$$

The item with the attractive exposure features but dissatisfying content will have a higher score of $Y_{u,e}$, which is then subtracted from the original prediction $Y_{u,i,e}$, lowering the rank of such items.

To summarize, compared to conventional recommender models, the proposed CR framework demonstrates three main differences:

- **Causal graph.** The recommender model under the CR framework is based on a new causal graph that accounts for the direct effect of exposure features on the prediction score.
- **Multi-task training.** In addition to the model learning in the real world (*i.e.*, $Y_{u,i,e}$), we also train the model to make predictions in the counterfactual world (*i.e.*, $Y_{u,i^*,e}$).
- **CR inference.** Instead of making recommendations according to the real-world prediction, we deduct the NDE of exposure features to mitigate the clickbait issue.

5 RELATED WORK

Recommendation. Because of the rich user/item features in the real-world scenarios [21, 22, 31], many approaches [7, 26] incorporate multi-modal user and item features into recommendation [6, 17, 23, 46]. Recently, Graph Neural Networks (GNN) [13, 14] have been widely used in recommendation [12, 48, 49], and GNN-based multi-modal model MMGCN [50] achieves promising performance due to its modality-aware information propagation over the user-item graph. However, existing works are trained by implicit feedback and totally ignore the clickbait issue. Therefore, items with many clicks but few likes will be recommended frequently.

Incorporating Various Feedback. To mitigate the clickbait issue, many efforts try to reduce the gap between clicks and likes by incorporating more features into recommendation, such as interaction context [25], item features [33], and various user feedback [53, 56]. Generally, they fall into two categories. The first is

negative experience identification [32, 58]. It performs a two-stage pipeline [32, 33] which first identifies negative interactions based on item features (e.g., the news quality) and context information (e.g., dwell time), and then only uses interactions with likes as positive samples. The second category considers directly incorporating extra post-click feedback (e.g., thumbs-up, favorite, and dwell time) to optimize recommender models [29, 53–55]. For instance, Wen *et al.* [51] leveraged the “skip” patterns to train recommender models with three kinds of items: “click-complete”, “click-skip”, and “non-click”. Nevertheless, the application of these methods is limited by the availability of context information and users’ additional post-click feedback. Post-click feedback is usually sparse, and thus using only clicks with likes for training will lose a large proportion of positive samples.

Causal Recommendation. In the information retrieval domain, early studies [2, 24] on causal inference mainly focus on de-biasing implicit feedback, e.g., position bias [9]. As to causal recommendation [5, 8, 60], many researchers study fairness [34] or the bias issues with the help of causal inference, such as exposure bias [3, 28] and popularity bias [1] in the logged data [30]. Among the family of causal inference for de-biasing recommendation, the most popular method is *Inverse Propensity Scoring Weighting* (IPW) [27, 41, 42] which turns the observed logged data into a pseudo-randomized trial by re-weighting samples. In general, they estimate the propensity of exposure or popularity at first, and re-weight samples with the inverse propensity scores. However, current causal recommendation never considers the clickbait issue. They don’t distinguish the effects of exposure and content features, and treat users’ implicit feedback such as clicks as the actual user preference. Therefore, prior studies still have the clickbait issue and recommend the items that many users would click but actually dislike.

6 EXPERIMENTS

6.1 Experimental Settings

Datasets. We evaluate our proposed CR framework on two publicly available datasets in different application scenarios: Tiktok [50] and Adressa [16]. For each dataset, we utilize post-click feedback to evaluate the recommender models. We admit that: 1) the sparsity of post-click feedback might restrict the scale of the evaluation, however, we still cover a large group of users for evaluation. Actually, almost all users are covered in two datasets; and 2) the items with more attractive exposure features are easier to be collected as testing samples regardless of content features since they are more likely to be clicked. Nevertheless, constructing a totally unbiased testing set is unrealistic without external intervention, which is extremely expensive and thus left to future work. The statistics of datasets are in Table 1.

- **Tiktok.** It is a multi-modal micro-video dataset released in ICME Challenge 2019³ where a micro-video has the features of caption, audio, and video. Multi-modal item features have already been extracted by the organizer for the fair comparison. We treat captions as exposure features and the remaining as content ones. Besides, actions of thumbs-up, favorite, or finish are used as

Table 1: Statistics of two datasets.

Dataset	#Users	#Items	#Clicks	#Likes
Tiktok	18,855	34,756	1,493,532	589,008
Adressa	31,123	4,895	1,437,540	998,612

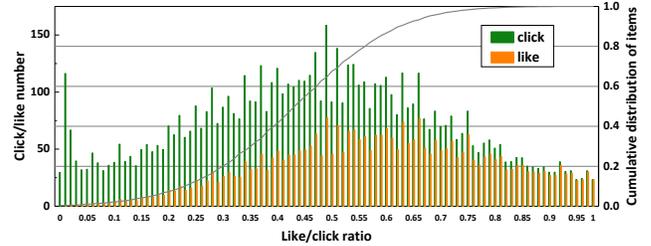


Figure 4: Click and like distributions of items in Tiktok. The grey line visualizes the cumulative proportion of items as the like/click ratio increases. The x-axis is like/click ratio and the y-axis is the number of clicks or likes.

the positive post-click feedback (i.e., like), which is only used to construct the testing set for evaluation.

- **Adressa**⁴. This is a news dataset [16] where the title and description of news are exposure features and the news content is treated as content features. We use the pre-trained Multilingual BERT [11] to extract textual features into 768-dimension vectors. Following prior studies [25], we treat a click with dwell time > 30 seconds as a like of user.

Figure 4 outlines the distribution of the like/click ratio where items are ranked and divided into 101 groups according to the ratio value. As can be seen, over 60% of items have like/click ratio smaller than 0.5, indicating the wide existence of clicks that end with dislikes. Moreover, recommending such items may lead to more clicks which fail to satisfy users and hurt user experience.

For each user, we randomly choose 10% clicks that end with likes to constitute a test set⁵, and treat the remaining as the training set. Besides, 10% of clicks are randomly selected from the training set as the validation set. We utilize the validation set to tune hyper-parameters and choose the best model for the testing phase. For each click, we randomly choose an item the user has never interacted with as the negative sample for training.

Evaluation Metrics. We follow the all-ranking evaluation protocol that ranks over all the items for each user except the clicked ones used in training [18, 47], and report the recommendation performance through: Precision@K (P@K), Recall@K (R@K) and NDCG@K (N@K) with $K = \{10, 20\}$ where higher values indicate better performance [50].

Compared Methods We compare the proposed CR with various recommender methods that might alleviate the clickbait issue. For a fair comparison, all methods are applied to MMGCN [50], which is the state-of-the-art multi-modal recommender model and captures the modality-aware high-order user-item relationships. Specifically, CR is compared with the following baselines:

- **NT.** Following [50], MMGCN is trained by the normal training (NT) strategy, where all item features are used and MMGCN is

⁴<http://reclab.idi.ntnu.no/dataset/>.

⁵If fewer than 10% clicks of a user end with likes, all such clicks are put into the test set. Besides, we ignore the potential noise in the test set, e.g., fake favorite.

³<http://ai-lab-challenge.bytedance.com/tce/vc/>.

Table 2: Top- K recommendation performance of compared methods on Tiktok and Adressa. %Improve. denotes the relative performance improvement of CR over NT. The best results are highlighted in bold. Stars and underlines denote the best results of the baselines with and without using additional post-click feedback during training, respectively.

Dataset Metric	Tiktok						Adressa					
	P@10	R@10	N@10	P@20	R@20	N@20	P@10	R@10	N@10	P@20	R@20	N@20
NT [50]	<u>0.0256</u>	<u>0.0357</u>	0.0333	<u>0.0231</u>	<u>0.0635</u>	0.0430	<u>0.0501</u>	<u>0.0975</u>	<u>0.0817</u>	<u>0.0415</u>	<u>0.1612</u>	<u>0.1059</u>
CFT [50]	0.0253	0.0356	<u>0.0339</u>	0.0226	0.0628	<u>0.0437</u>	0.0482	0.0942	0.0780	0.0405	0.1573	0.1021
IPW [27]	0.0230	0.0334	0.0314	0.0210	0.0582	0.0406	0.0419	0.0804	0.0663	0.0361	0.1378	0.0883
CT [50]	0.0217	0.0295	0.0294	0.0194	0.0520	0.0372	0.0493	0.0951	0.0799	0.0418*	0.1611	0.1051
NR [51]	0.0239	0.0346	0.0329	0.0216	0.0605	0.0424	0.0499	0.0970	0.0814	0.0415	0.1610	0.1058
RR	0.0264*	0.0383*	0.0367*	0.0231*	0.0635*	0.0430*	0.0521*	0.1007*	0.0831*	0.0415	0.1612*	0.1059*
CR	0.0269	0.0393	0.0370	0.0242	0.0683	0.0476	0.0532	0.1045	0.0878	0.0439	0.1712	0.1133
%Improve.	5.08%	10.08%	11.11%	4.76%	7.56%	10.70%	6.19%	7.18%	7.47%	5.78%	6.20%	6.99%

optimized with click data. We keep the same hyperparameter settings as in [50], including that: the model is optimized by the BPR loss [40]; the learning rate is set as 0.001, and the size of latent features is 64.

- **CFT.** Based on the analysis that exposure features are easy to induce the clickbait issue, we only use content features for training (CFT). The model is also trained with all click data.
- **IPW.** Liang *et al.* [27, 28] tried to reduce the exposure bias from clicks by causal inference with IPW [41]. For a fair comparison, we follow the idea of Liang *et al.* and implement the exposure and click models in [27] by MMGCN since it uses multi-modal item features and thus can achieve better performance.

Besides, considering post-click feedback can indicate the actual user satisfaction, we compare CR with three baselines that additionally incorporate post-click feedback:

- **CT.** This method is conducted in the clean training (CT) setting, in which only the clicks that end with likes are viewed as positive samples to train MMGCN.
- **NR.** Wen *et al.* [51] adopted post-click feedback and also treated “click-skip” items as negative samples. We apply their Negative feedback Re-weighting (NR) into MMGCN. In detail, NR adjusts the weights of two negative samples during training, including “click-skip” items and “no-click” items. Following [51], the extra hyper-parameter $\lambda_{p,n}$, *i.e.*, the ratio of two kinds of negative samples, is tuned in $\{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$.
- **RR.** For each user, we propose a strategy to re-rank (RR) the top 20 items recommended by NT during inference. For each item, the final ranking is calculated by the sum of rank in NT and the rank based on the like/click ratio of items. The like/click ratio is calculated from the whole dataset.

We omit potential testing recommender models such as VBPR [17] since the previous work [50] has validated the superior performance of MMGCN over these multi-modal recommender models.

Parameter Settings. We strictly follow the original implementation of MMGCN [50], including code, parameter initialization, and hyperparameter tuning. The additional weight α in the multi-task loss function is tuned in $\{0, 0.25, 0.5, 0.75, 1, 2, 3, 4, 5\}$. The effect of α on the performance is visualized in Figure 5 where the model obtains the best performance when α is 1 or 2, showing the effectiveness of our proposed multi-task training. As shown in Table

Table 3: Results of estimating NDE under $T = t$.

Method	Tiktok		Adressa	
	R@20	N@20	R@20	N@20
NT	0.0635	0.0430	0.1612	0.1059
CR ($T = t$)	0.0671	0.0465	0.1667	0.1093
CR ($T = t^*$)	0.0683	0.0476	0.1712	0.1133

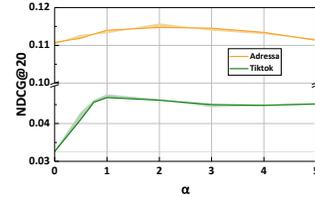


Figure 5: Effect of α in the multi-task loss.

3, we estimate the NDE of $E = e$ on Y under situation $T = t^*$ due to its rationality and better performance. Moreover, early stopping is performed for the model selection, *i.e.*, stop training if recall@10 on the validation set does not increase for 10 successive epochs. We train all the models multiple times and report the average performance. More details can be found in the code⁶.

6.2 Performance Comparison

The overall performance comparison is summarized in Table 2. From the table, we have the following observations:

- **Debiasing Training.** In most cases, CFT performs worse than NT, which is attributed to discarding exposure features. The result overrules the option of simply discarding exposure features to mitigate the clickbait issue, which is indispensable for user preference prediction. Moreover, the performance of IPW is inferior on Tiktok and Adressa, showing that the clickbait issue may not be resolved by simply discouraging the recommendation of items with more clicks. In addition, the result indicates the importance of accurate propensity estimation to mitigate a bias, which is the crucial barrier of the usage of IPW for handling the bias caused by features with complex and changeable patterns.

- **Post-click Feedback.** RR outperforms NT, which re-ranks the recommendations of NT according to the like/click ratio. It validates the effectiveness of leveraging post-click feedback to mitigate the clickbait issue and satisfy user requirements. However, CT and

⁶<https://github.com/WenjieWWJ/Clickbait/>.

Table 4: Performance comparison between CR inference and the inference via TE.

Dataset Metric	Tiktok			Adressa		
	P@20	R@20	N@20	P@20	R@20	N@20
NT	0.0231	0.0635	0.0430	0.0415	0.1612	0.1059
CR-TE	0.0235	0.0665	0.0461	0.0436	0.1698	0.1122
CR inference	0.0242	0.0683	0.0476	0.0439	0.1712	0.1133

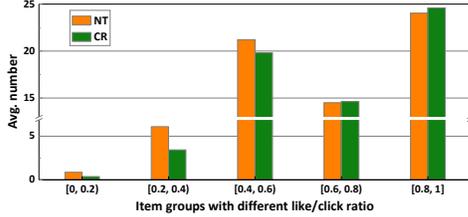


Figure 6: Visualization of the averaged recommendation frequencies of items. Note that items with low like/click ratios shouldn't be recommended.

NR, which incorporate post-click feedback into the model training, perform worse than NT on Tiktok, *e.g.*, the NDCG@10 of CT decreases by 11.71% on Tiktok. We ascribe the inferior performance to the sparsity of post-click feedback, which hurts the model generalization when the model is trained on a small number of interactions. It makes sense since the clicks that end with likes in Tiktok account for only 39.44%, which is much lower than that in Adressa (69.47%). Moreover, we postulate the reason to be the inaccurate causal graph (Figure 3(a)) that lacks the direct edge from exposure features to prediction, which is further detailed in Table 4.

• **CR Inference.** In all cases, CR achieves significant performance gains over all baselines. In particular, CR outperforms NT *w.r.t.* N@10 by 11.11% and 7.47% on Tiktok and Adressa, respectively. The result validates the effectiveness of the proposed CR, which is attributed to the new causal graph and counterfactual inference. In particular, CR also outperforms RR which additionally considers the post-click feedback. This further signifies the rationality of CR in eliminating the direct effect of exposure features on the prediction to mitigate the clickbait issue. As such, CR significantly helps to recommend more satisfying items, which can improve the user engagement and produce greater economic benefits.

6.2.1 Effect of the Proposed Causal Graph. To shed light on the performance gain, we further study one variant, *i.e.*, CR-TE, which performs inference via the TE of $E = e$ and $T = t$, *i.e.*, its difference from NT is training over the proposed causal graph. Table 4 shows their performance with $K = 20$. From the table, we observe that CR-TE outperforms NT, which justifies the rationality of incorporating the direct edge from exposure features to the prediction score. It validates the existence of the shortcut where exposure features can directly lead to clicks. Moreover, CR inference further outperforms CR-TE, showing that reducing the direct effect of exposure features indeed mitigates the clickbait issue and leads to better recommendation with more satisfaction.

6.3 In-depth Analysis

We then take CR on Adressa as an example to further investigate the effectiveness of CR.

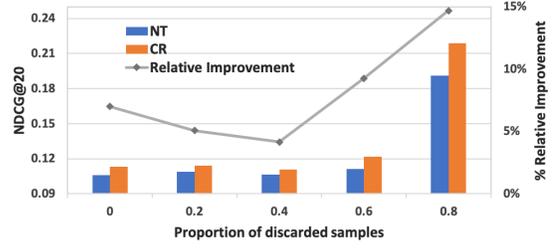


Figure 7: Performance comparison across the subsets of Adressa with different discarding proportions. A larger proportion indicates a higher percentage of the clicks that end with dislikes in the dataset.

6.3.1 Visualization of Recommendations *w.r.t.* Like/click Ratio. Recall that recommender models with the clickbait issue tend to recommend items even though their like/click ratios are low. We thus compare the recommendations of CR and NT to explore whether CR can reduce recommending the items with high risk to hurt user experience. Specifically, we collect top-ranked items recommended to each user and count the frequency of each item being recommended. Figure 6 outlines the recommendation frequencies of CR and NT where items are intuitively split into five groups according to their like/click ratio for better visualization. From the figure, we can see that as compared to NT, 1) CR recommends fewer items with like/click ratios ≤ 0.6 ; and 2) more items with high like/click ratios, especially in $[0.8, 1]$. The result indicates the higher potential of CR to satisfy users, which is attributed to the proper modeling of the effect of exposure features.

6.3.2 Effect of Dataset Cleaness. We then study how the effectiveness of CR is influenced by the “cleanness” of the click data. Specifically, we compare CR and NT over filtered datasets with different percentages of clicks that end with dislikes. We rank the items in descending order by the like/click ratio, and discard the top-ranked items at a certain proportion where a larger discarding proportion leads to a dataset with a higher percentage of clicks that end with dislikes. Figure 7 shows the performance with discarding proportion changing from 0 (the original dataset) to 0.8. From Figure 7, we have the following findings: 1) CR outperforms NT in all cases, which further validates the effectiveness of CR. 2) The performance gains are close when the discarding proportion is smaller than 0.4, and increase dramatically under larger proportions. The result indicates that mitigating the clickbait issue is more important for the recommendation scenarios with more clicks that end with dislikes.

6.3.3 Effect of Fusion Strategy. Recall that any differentiable arithmetic binary operations can be equipped as the fusion strategy in CR [35]. To shed light on the development of proper fusion strategies, we investigate its essential properties, such as linearity and boundary. As such, in addition to the MUL strategy, we further evaluate a vanilla SUM strategy with linear fusion, SUM with sigmoid function, and SUM/MUL with $\tanh(\cdot)$ as the activation function. Formally,

$$\begin{cases} \text{SUM-linear: } Y_{u,i,e} = f(Y_{u,i}, Y_{u,e}) = Y_{u,i} + Y_{u,e}, \\ \text{SUM-sigmoid: } Y_{u,i,e} = f(Y_{u,i}, Y_{u,e}) = Y_{u,i} + \sigma(Y_{u,e}), \\ \text{SUM-tanh: } Y_{u,i,e} = f(Y_{u,i}, Y_{u,e}) = Y_{u,i} + \tanh(Y_{u,e}), \\ \text{MUL-tanh: } Y_{u,i,e} = f(Y_{u,i}, Y_{u,e}) = Y_{u,i} * \tanh(Y_{u,e}). \end{cases} \quad (12)$$

Table 5: Performance of CR with different fusion strategies.

Metric	P@10	R@10	N@10	P@20	R@20	N@20
SUM-Linear	0.0380	0.0718	0.0598	0.0317	0.1196	0.0780
SUM-tanh	0.0537	0.1060	0.0889	0.0447	0.1744	0.1150
MUL-tanh	0.0520	0.1027	0.0861	0.0435	0.1698	0.1118
SUM-sigmoid	0.0533	0.1044	0.0877	0.0439	0.1714	0.1132
MUL-sigmoid	0.0532	0.1045	0.0878	0.0439	0.1711	0.1132

Similar to the MUL fusion strategy, we also estimate CR inference for SUM-linear, SUM-sigmoid, SUM-tanh, and MUL-tanh, respectively. The results are as follows:

$$\begin{cases} \text{SUM-linear: } Y_{CR} = Y_{u,i} - c_{u,i} \propto Y_{u,i}, \\ \text{SUM-sigmoid: } Y_{CR} = Y_{u,i} - c_{u,i} \propto Y_{u,i}, \\ \text{SUM-tanh: } Y_{CR} = Y_{u,i} - c_{u,i} \propto Y_{u,i}, \\ \text{MUL-tanh: } Y_{CR} = (Y_{u,i} - c_{u,i}) * \tanh(Y_{u,e}). \end{cases}$$

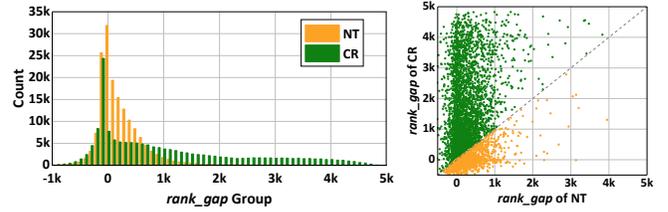
During CR inference, the SUM strategies with different activation functions are equivalent. However, they capture the direct effect of exposure features differently in the training process. Therefore, the recommendation results are theoretically different.

The performance of different fusion strategies is reported in Table 5. From that, we can find that: 1) non-linear fusion strategies are significantly better than linear ones due to the better representation capacity; and 2) SUM-tanh achieves the best performance over the other fusion strategies, including the proposed MUL-sigmoid strategy. This shows that a fusion function with the proper boundary can further improve the performance of CR and multiple fusion strategies are worth studying when CR inference is applied to other datasets in future.

6.3.4 CR Evaluation on Synthetic Data. To further evaluate the effectiveness of CR on mitigating the direct effect of exposure features, we conduct experiments on synthetic data. Specifically, during inference, we construct a fake item for each positive user-item pair in the testing data by “poisoning” the exposure feature of the item. The content features of the fake item are the same as the real item while its exposure features are randomly selected from the items with the like/click ratio < 0.5 . Such items with low like/click ratios are more likely to be the ones with the clickbait issue. Their exposure features are easy to be attractive but deceptive, for example, “Find UFO!”. Besides, there is a large discrepancy between the exposure and content features of the fake items, which simulates the items with the clickbait issue where content features do not align with exposure features. Therefore, the fake item should have a lower rank than the paired real item if the recommender model can mitigate the clickbait issue well.

A lower rank of the fake item indicates a better elimination of the direct effect from the exposure features. Accordingly, we rank all testing real items and the fakes ones for each user, and we define $rank_gap = rank_{fake} - rank_{real}$ to measure the performance of recommender models, where $rank_{fake}$ and $rank_{real}$ are the ranks of the paired fake and real items, respectively. A larger $rank_gap$ value indicates a bigger gap and thus better performance. Lastly, we calculate the $rank_gap$ of each triplet $\langle \text{user, real item, fake item} \rangle$ in the testing data.

As shown in Figure 8(a), the $rank_gap$ values are first grouped, and then counted by group. From this figure, we can observe that the $rank_gap$ values generated by CR are larger than those of NT, and



(a) Distribution w.r.t. $rank_gap$ group.

(b) $rank_gap$ of NT and CR.

Figure 8: Results of CR evaluation on synthetic data.

the distribution of CR is flatter than that of NT, indicating that CR produces lower ranking scores for the fake items. This is because CR effectively reduces the direct effect of deceptive exposure features. Besides, we randomly sample 5k samples of triplets from the testing data and individually compare the $rank_gap$ values generated by CR and NT in Figure 8(b). From the figure, we can find that 1) most points are above the diagonal, showing the $rank_gap$ of CR is usually larger than that of NT; and 2) the $rank_gap$ values generated by CR cover a wider range, varying from 0 to 5k. The findings imply that CR can distinguish the real and fake items well, which further proves the effectiveness of CR on mitigating the clickbait issue.

7 CONCLUSION AND FUTURE WORK

The clickbait issue widely exists in the industrial recommender systems. To eliminate its effect, we proposed a new recommendation framework CR that accounts for the causal relations among the exposure features, content features, and predictions. Through performing counterfactual inference, we estimated the direct effect of exposure features on the prediction and removed it from recommendation scoring. While we instantiated CR on a specific recommender model MMGCN, it is model-agnostic and only requires minor adjustments (several lines of codes) to be adopted to other models, enabling the wide usage of CR across different recommendation scenarios and models. By mitigating the clickbait issue, they can improve the user satisfaction and engagement.

This work opens up a new research direction—incorporating counterfactual inference into recommender systems. Following this direction, there are many interesting ideas that deserve our exploration. 1) Considering the huge benefit of reasoning over causal graph, it is essential to construct a more comprehensive causal graph for recommendation with more fine-grained causal relations in future. 2) This work justifies the effectiveness of counterfactual inference on mitigating the clickbait issue, and motivates further exploration on other intrinsic biases and issues in the click data, such as selection bias [36] and position bias [24]. 3) More broadly, this work signifies the importance of causal inference on recommendation. It opens the door of empowering recommender systems with more causal inference techniques, such as intervention and counterfactual inference.

ACKNOWLEDGMENTS

This research/project is supported by the Sea-NExT Joint Lab, the National Natural Science Foundation of China (U19A2079) and National Key Research and Development Program of China (2020AAA0106000).

REFERENCES

- [1] Himan Abdollahpour, Robin Burke, and Bamshad Mobasher. 2019. Managing popularity bias in recommender systems with personalized re-ranking. In *Proceedings of the International Flairs Conference*. AAAI Press.
- [2] Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W. Bruce Croft. 2018. Unbiased Learning to Rank with Unbiased Propensity Estimation. In *SIGIR*. ACM, 385–394.
- [3] Stephen Bonner and Flavian Vasile. 2018. Causal embeddings for recommendation. In *RecSys*. ACM, 104–112.
- [4] Remi Cadene, Corentin Dancette, Hedi Ben-younes, Matthieu Cord, Devi Parikh, et al. 2019. Rubi: Reducing unimodal biases for visual question answering. In *NeuIPS*. 841–852.
- [5] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and Debias in Recommender System: A Survey and Future Directions. *arXiv preprint arXiv:2010.03240* (2020).
- [6] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *SIGIR*. ACM, 335–344.
- [7] Xusong Chen, Dong Liu, Zheng-Jun Zha, Wengang Zhou, Zhiwei Xiong, and Yan Li. 2018. Temporal Hierarchical Attention at Category- and Item-Level for Micro-Video Click-Through Prediction. In *MM*. ACM, 1146–1153.
- [8] Konstantina Christakopoulou, Madeleine Traverse, Trevor Potter, Emma Marriott, Daniel Li, Chris Haulk, Ed H. Chi, and Minmin Chen. 2020. Deconfounding User Satisfaction Estimation from Response Rate Bias. In *RecSys*. ACM, 450–455.
- [9] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. 2008. An Experimental Comparison of Click Position-Bias Models. In *WSDM*. ACM, 87–94.
- [10] Balázs Csánád Csáji. 2001. Approximation with artificial neural networks. *Faculty of Sciences, Eötvös Loránd University, Hungary* 24, 48 (2001), 7.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *arXiv:1810.04805*.
- [12] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. In *WWW*. ACM, 417–426.
- [13] Fuli Feng, Xiangnan He, Yiqun Liu, Liqiang Nie, and Tat-Seng Chua. 2018. Learning on partial-order hypergraphs. In *WWW*. ACM, 1523–1532.
- [14] Fuli Feng, Xiangnan He, Jie Tang, and Tat-Seng Chua. 2019. Graph adversarial training: Dynamically regularizing based on graph structure. *TKDE* (2019).
- [15] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
- [16] Jon Atle Gulla, Lemei Zhang, Peng Liu, Özlem Özgöbek, and Xiaomeng Su. 2017. The Adressa Dataset for News Recommendation. In *WI*. ACM, 1042–1048.
- [17] Ruining He and Julian McAuley. 2016. VBPR: visual bayesian personalized ranking from implicit feedback. In *AAAI*. AAAI press.
- [18] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. ACM, 639–648.
- [19] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. ACM, 173–182.
- [20] Katja Hofmann, Fritz Behr, and Filip Radlinski. 2012. On Caption Bias in Interleaving Experiments. In *CIKM*. ACM, 115–124.
- [21] Richang Hong, Lei Li, Junjie Cai, Dapeng Tao, Meng Wang, and Qi Tian. 2017. Coherent semantic-visual indexing for large-scale image retrieval in the cloud. *TIP* 26, 9 (2017), 4128–4138.
- [22] Richang Hong, Yang Yang, Meng Wang, and Xian-Sheng Hua. 2015. Learning visual semantic relationships for efficient visual retrieval. *Transactions on Big Data* 1, 4 (2015), 152–161.
- [23] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-scale Time-aware User Interest Modeling for Micro-video Recommendation. In *MM*. ACM, 3487–3495.
- [24] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*. ACM, 781–789.
- [25] Youngho Kim, Ahmed Hassan, Ryen W White, and Imed Zitouni. 2014. Modeling dwell time to predict click-level satisfaction. In *WSDM*. ACM, 193–202.
- [26] Yongqi Li, Meng Liu, Jianhua Yin, Chaoran Cui, Xin-Shun Xu, and Liqiang Nie. 2019. Routing micro-videos via a temporal graph-guided recommendation system. In *MM*. ACM, 1464–1472.
- [27] Dawen Liang, Laurent Charlin, and David M Blei. 2016. Causal inference for recommendation. In *UAI*. AUAI.
- [28] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *WWW*. ACM, 951–961.
- [29] Chao Liu, Ryen W White, and Susan Dumais. 2010. Understanding web browsing behaviors through Weibull analysis of dwell time. In *SIGIR*. ACM, 379–386.
- [30] Dugang Liu, Pengxiang Cheng, Zhenhua Dong, Xiuqiang He, Weike Pan, and Zhong Ming. 2020. A General Knowledge Distillation Framework for Counterfactual Recommendation via Uniform Data. In *SIGIR*. ACM, 831–840.
- [31] Meng Liu, Leigang Qu, Liqiang Nie, Maofu Liu, Lingyu Duan, and Baoquan Chen. 2020. Iterative Local-Global Collaboration Learning Towards One-Shot Video Person Re-Identification. *TIP* 29 (2020), 9360–9372.
- [32] Hongyu Lu, Min Zhang, and Shaoping Ma. 2018. Between Clicks and Satisfaction: Study on Multi-Phase User Preferences and Satisfaction for Online News Reading. In *SIGIR*. ACM, 435–444.
- [33] Hongyu Lu, Min Zhang, Weizhi Ma, Ce Wang, Feng xia, Yiqun Liu, Leyu Lin, and Shaoping Ma. 2019. Effects of User Negative Experience in Mobile News Streaming. In *SIGIR*. ACM, 705–714.
- [34] Marco Morik, Ashudeep Singh, Jessica Hong, and Thorsten Joachims. 2020. Controlling Fairness and Bias in Dynamic Learning-to-Rank. In *SIGIR*. ACM, 429–438.
- [35] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. 2020. Counterfactual VQA: A Cause-Effect Look at Language Bias. In *arXiv:2006.04315*.
- [36] Zohreh Ovaisi, Ragib Ahsan, Yifan Zhang, Kathryn Vasilaky, and Elena Zheleva. 2020. Correcting for Selection Bias in Learning-to-Rank Systems. In *WWW*. ACM, 1863–1873.
- [37] Judea Pearl. 2001. Direct and indirect effects. In *UAI*. Morgan Kaufmann Publishers Inc, 411–420.
- [38] Judea Pearl. 2009. *Causality*. Cambridge university press.
- [39] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect* (1st ed.). Basic Books, Inc.
- [40] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press, 452–461.
- [41] Paul R. Rosenbaum and Donald B. Rubin. 1983. The central role of the propensity score in observational studies for causal effects. *Biometrika* 70, 1 (04 1983), 41–55.
- [42] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*. ACM, 501–509.
- [43] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. 2020. Long-Tailed Classification by Keeping the Good and Removing the Bad Momentum Causal Effect. In *NeurIPS*.
- [44] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. 2020. Unbiased scene graph generation from biased training. In *arXiv:2002.11949*.
- [45] Tyler J VanderWeele. 2013. A three-way decomposition of a total effect into direct, indirect, and interactive effects. *Epidemiology (Cambridge, Mass.)* 24, 2 (2013), 224.
- [46] Wenjie Wang, Ling-Yu Duan, Hao Jiang, Peiguang Jing, Xuemeng Song, and Liqiang Nie. 2021. Market2Dish: Health-Aware Food Recommendation. *TOMM* 17 (April 2021).
- [47] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. ACM, 373–381.
- [48] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *SIGIR*. ACM, 165–174.
- [49] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. 2020. Graph-Refined Convolutional Network for Multimedia Recommendation with Implicit Feedback. In *MM*. 3541–3549.
- [50] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal Graph Convolution Network for Personalized Recommendation of Micro-video. In *MM*. ACM, 1437–1445.
- [51] Hongyi Wen, Longqi Yang, and Deborah Estrin. 2019. Leveraging Post-click Feedback for Content Recommendations. In *RecSys*. ACM, 278–286.
- [52] Hong Wen, Jing Zhang, Yuan Wang, Fuyu Lv, Wentian Bao, Quan Lin, and Keping Yang. 2020. Entire space multi-task modeling via post-click behavior decomposition for conversion rate prediction. In *SIGIR*. ACM, 2377–2386.
- [53] Byoungju Yang, Sangkeun Lee, Sungchan Park, and Sang goo Lee. 2012. Exploiting Various Implicit Feedback for Collaborative Filtering. In *WWW*. ACM, 639–640.
- [54] Xing Yi, Liangjie Hong, Erheng Zhong, Nanthan Nan Liu, and Suju Rajan. 2014. Beyond clicks: dwell time for personalization. In *RecSys*. ACM, 113–120.
- [55] Peifeng Yin, Ping Luo, Wang-Chien Lee, and Min Wang. 2013. Silence is Also Evidence: Interpreting Dwell Time for Recommendation from Psychological Perspective. In *KDD*. ACM, 989–997.
- [56] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient transfer from sequential behaviors for user modeling and recommendation. In *SIGIR*. ACM, 1469–1478.
- [57] Yisong Yue, Rajan Patel, and Hein Roehrig. 2010. Beyond Position Bias: Examining Result Attractiveness as a Source of Presentation Bias in Clickthrough Data. In *WWW*. ACM, 1011–1018.
- [58] Savvas Zannettou, Sotirios Chatzis, Kostantinos Papadamou, and Michael Sirivianos. 2018. The good, the bad and the bait: Detecting and characterizing clickbait on YouTube. In *SPW*. IEEE, 63–69.
- [59] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-Scale Causal Approaches to Debiasing Post-Click Conversion Rate Estimation with Multi-Task Learning. In *WWW*. ACM, 2775–2781.
- [60] Ziwei Zhu, Yun He, Yin Zhang, and James Caverlee. 2020. Unbiased Implicit Recommendation and Propensity Estimation via Combinational Joint Learning. In *RecSys*. ACM, 551–556.