



Dynamic Causal Collaborative Filtering

Shuyuan Xu

Rutgers University

New Brunswick, NJ, US

shuyuan.xu@rutgers.edu

Juntao Tan

Rutgers University

New Brunswick, NJ, US

juntao.tan@rutgers.edu

Zuohui Fu

Rutgers University

New Brunswick, NJ, US

zuohui.fu@rutgers.edu

Jianchao Ji

Rutgers University

New Brunswick, NJ, US

jianchao.ji@rutgers.edu

Shelby Heinecke

Salesforce Research

Palo Alto, CA, US

shelby.heinecke@salesforce.com

Yongfeng Zhang

Rutgers University

New Brunswick, NJ, US

yongfeng.zhang@rutgers.edu

ABSTRACT

Causal graph, as an effective and powerful tool for causal modeling, is usually assumed as a Directed Acyclic Graph (DAG). However, recommender systems usually involve feedback loops, defined as the cyclic process of recommending items, incorporating user feedback in model updates, and repeating the procedure. As a result, it is important to incorporate loops into the causal graphs to accurately model the dynamic and iterative data generation process for recommender systems. However, feedback loops are not always beneficial since over time they may encourage more and more narrowed content exposure, which if left unattended, may result in echo chambers. As a result, it is important to understand when the recommendations will lead to echo chambers and how to mitigate echo chambers without hurting the recommendation performance.

In this paper, we design a causal graph with loops to describe the dynamic process of recommendation. We then take Markov process to analyze the mathematical properties of echo chamber such as the conditions that lead to echo chambers. Inspired by the theoretical analysis, we propose a Dynamic Causal Collaborative Filtering (∂ CCF) model, which estimates users' post-intervention preference on items based on back-door adjustment and mitigates echo chamber with counterfactual reasoning. Multiple experiments are conducted on real-world datasets and results show that our framework can mitigate echo chambers better than other state-of-the-art frameworks while achieving comparable recommendation performance with the base recommendation models.

CCS CONCEPTS

• Computing methodologies → Machine learning; • Information systems → Recommender systems.

KEYWORDS

Collaborative Filtering; Causal Machine Learning; Counterfactual Reasoning; Recommender Systems; Echo Chambers

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 the author(s) 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.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-9236-5/22/10...\$15.00

<https://doi.org/10.1145/3511808.3557300>

ACM Reference Format:

Shuyuan Xu, Juntao Tan, Zuohui Fu, Jianchao Ji, Shelby Heinecke, and Yongfeng Zhang. 2022. Dynamic Causal Collaborative Filtering. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3511808.3557300>

1 INTRODUCTION

Recommender Systems (RS) aim to provide personalized services for users, occupying an expanding role in a wide range of applications, such as e-commerce, video streaming, social media and online job markets. Recent efforts have been made towards causal recommendation, which incorporates causation into recommender systems to enable causal inference over some critical aspects, such as eliminating bias [5, 44], promoting fairness [14, 30, 59], improving robustness [28] and enhancing explainability [16, 48, 50].

Causal graph is an effective and powerful tool that enables researchers to estimate desired values [17, 55]. Typically, the causal graph is constructed as a directed acyclic graph (DAG) depicting the data generation process. However, in real-word recommender systems, the data generation process usually spans over a period of time. Within this period, the recommendation results made by the system can have a great impact over users' interests and decision preference, and in turn influence the feedback that the system receives. This dynamic process is called a *feedback loop* [15, 21], which may not be accurately captured by a DAG causal graph. Therefore, introducing loops into the causal graph design may allow causal models to understand the dynamic data generation process in recommender systems more comprehensively.

While the causal graph with loops can better capture the dynamic and iterative data generation process in recommender systems, the presence of feedback loops is not always beneficial. Specifically, feedback loops may narrow the user's interest towards certain contents, which may further lead to decreased engagement with the system [7]. Additionally, feedback loops, if left unattended, may also result in what is known as *echo chambers* [7, 15]. In general, echo chambers describe the homogenization of social communities [46] that occur as a result of feedback loops. This homogenization isolates users in information echo chambers, severely limiting their information exposure. The existence of echo chambers has been validated on recommender systems with respect to e-commerce [15], online media [1, 20, 34], and social networking [39, 45].

Take movie recommendation as an example to better understand the consequences of echo chambers. Suppose a user has provided

positive feedback towards action movies, and the system is able to learn that the user would be highly possible to be interested in action movies. If the system does not properly handle echo chamber when making recommendations, it may repeatedly recommend other action movies and will likely receive positive feedback in the short term. However, over time the system will gradually converge to only showing action movies and make the user dissatisfied eventually. Given the ubiquity of recommender systems in our daily-life, it is important to understand when a recommender system results in echo chambers and how to mitigate echo chambers while maintaining the recommendation performance.

In this paper, we particularly seek to answer the following three questions regarding the task of mitigating echo chambers: 1) how to design a causal graph with loops to describe the dynamic and iterative data generation process of recommendation, 2) if a recommender system does not take care of echo chamber, when will the system result in echo chamber, and 3) how to mitigate echo chambers without hurting the recommendation performance.

Specifically, we first design a causal graph as a directed graph with loops to describe how the data is generated in a dynamic and iterative manner. Based on our constructed causal graph, we conduct mathematical analysis to understand the conditions that a system will result in echo chamber. Concretely, we represent the dynamic recommendation process as a Markov Process. Then we categorize user behavior into three types and check the existence of homogeneity brought by echo chambers. The analysis confirms the soundness of our proposed causal graph (Section 3.2).

In addition to the above theoretical contributions, our work also provides essential technical contributions. We propose a Dynamic Causal Collaborative Filtering (∂ CCF) framework for recommendation to mitigate echo chambers. More specifically, we apply the back-door adjustment to estimate the post-intervention effect based on the unfolded causal graph. Inspired by our theoretical analysis, we apply counterfactual reasoning to mitigate the echo chamber effect while retaining the recommendation performance (Section 4). We conduct experiments over two real-world datasets to evaluate the effect of echo chambers by measuring the change of content diversity [15]. The results confirm that our framework is capable of mitigating echo chambers while obtaining comparable recommendation performance with the base recommendation models.

The key contributions of our paper can be summarized as follows:

- We design a causal graph with loops to represent the dynamic data generation process of recommender systems.
- We represent the user-system interaction as a Markov Process to understand the conditions that lead a system to echo chambers if the system does not properly handle echo chamber.
- We propose Dynamic Causal Collaborative Filtering (∂ CCF), which takes the back-door adjustment to estimate user preferences and applies counterfactual reasoning to mitigate echo chambers.
- Experiments on two real-world datasets show that our framework can mitigate echo chambers better than other methods while maintaining comparable recommendation performance with the base recommendation models.

The remainder of this paper is organized as follows. We discuss the related work in Section 2. In Section 3, we introduce our designed causal graph and the theoretical analysis of echo chambers.

We show the details of our proposed dynamic causal collaborative filtering framework in Section 4. We describe our experiments on real-world datasets and discuss the results in Section 5. Finally, in Section 6, we conclude the work and discuss future directions.

2 RELATED WORK

2.1 Causal Recommendation

Causal machine learning have been explored to tackle some critical problems in recommender systems. For example, researchers have leveraged causal models to enhance explainability [16, 47, 48, 50, 62], promote fairness [14, 29, 30, 59], eliminate bias [5, 31, 38, 44, 51, 56, 57, 64], improve robustness [13, 28], estimate the recommendation uplifts [41–43], and enable counterfactual reasoning for data augmentation [52, 54, 58, 60]. Across the various causal recommendation models, causal graph is shown to be a powerful tool that enables counterfactual reasoning. Existing models usually assume that causal graphs are directed acyclic graphs (DAG). However, the data used to train recommender systems is generated in a continuous and dynamic manner, which cannot be captured by a DAG as a single snapshot. Therefore, a causal graph with loops is needed to describe the dynamic recommendation scenario.

2.2 Echo Chambers

Echo chambers have been studied under several contexts such as social networks [3, 9, 12, 39, 45] and opinion dynamics [4, 26, 36]. Meanwhile, the importance of content diversity has been explored to address user polarization problems [2, 23, 27, 32, 34]. In recent years, the study of feedback loops and echo chambers in recommender systems has received a lot of attention [7, 15, 21, 33]. For example, Ge et al. [15] analyzed the property of echo chambers in e-commerce recommender systems. Existing works have demonstrated preliminary progress toward addressing feedback loops and mitigating echo chambers. For example, Jiang et al. [21] use the dynamic system framework to model user interest and treat interest extremes as a degeneracy point of the system. However, they assume that users and items are independent of each other, which is inconsistent with the principle of how collaborative filtering works. Kalimeris et al. [22] study the similar phenomenon and solve it by learning a stable fixed point, where user preferences do not change in response to the system recommendations. However, this work is limited to matrix factorization-based recommendation and does not consider causality. Instead, our work aims to mitigate echo chambers in a more general setting based on causal reasoning.

3 THEORETICAL ANALYSIS

In this section, we introduce our designed causal graph and the theoretical analysis of under what conditions will recommender systems result in echo chamber.

3.1 The Causal Graph

Figure 1(a) shows our designed causal graph for dynamic recommendation. Existing works usually assume the causal graph as a directed acyclic graph (DAG). However, as we mentioned before, a causal graph with loops may better capture the mechanism in recommender systems. We use uppercase letters (e.g., U) to denote

random variables and lowercase letters (e.g., u) to denote the corresponding specific values. We explain the rationality of our designed causal graph from the view of data generation as follows:

- Node U represents the user variable. More specifically, we take user ID as the variable values.
- Node V represents the exposed item variable. Similarly, we take item ID as the variable values.
- Node Y represents preference score. In this work, for simplicity, we consider variable Y as a binary variable, i.e., $Y = 1$ for *like* and $Y = 0$ for *dislike*.
- Node X represents the user interaction history. More specifically, it is a sequence of item IDs and the corresponding preference scores from the user.
- Edges $\{U, X\} \rightarrow V$ denote that the exposed item V is determined by user U and the user interaction history X .
- Edges $\{U, X, V, Y\} \rightarrow X$ denote the dynamic generation process of the user interaction history X . When an exposed item $V = v$ is generated, user's preference towards v as $Y = y_v$, combined with the exposed item v will be integrated into the history and update the previous history as a new history X , which is used to estimate the next exposed item. These edges representing the iterative and dynamic process construct the feedback loop in our designed causal graph.
- Edges $\{U, X, V\} \rightarrow Y$ denote that the user preference score is determined by user U , user interaction history X and the exposed item V . The preference of a user-item pair is not only determined by the corresponding (u, v) pair but also affected by the previous interaction history. Recall the example in Section 1, for a user who is interested in action movies, it is highly likely to obtain a positive feedback for an action movie in the beginning, but if the user has already watched so many action movies, the user may be tired of action movies and provide a negative feedback towards a new one.

From the causal graph, we can see four loops $X \rightarrow X$, $X \rightarrow V \rightarrow X$, $X \rightarrow Y \rightarrow X$ and $X \rightarrow V \rightarrow Y \rightarrow X$. These loops represent the dynamic and iterative updating process of the user interaction history, embodying the feedback loop in recommender systems.

3.2 Understanding Echo Chambers

In this section, we will provide a theoretical analysis of echo chambers in recommender systems to understand under what condition the system will lead to echo chambers if the system does not properly handle echo chambers. The following analysis targets recommendation models that do not consider echo chambers.

3.2.1 Problem Setting and Notations. Generally speaking, recommender systems, especially Collaborative Filtering (CF) based models, explicitly or implicitly learn the similarity between items for recommendation [10, 61, 62]. For example, Matrix Factorization (MF) models such as [24] predict $\mathbf{u}^\top \mathbf{v}$ for recommendation, where \mathbf{u} and \mathbf{v} are the user and item latent factors, and such models learn the similarity between items that are projected by user latent factor \mathbf{u} (i.e., for a certain user u , if $\mathbf{u}^\top \mathbf{v}_1$ is similar to $\mathbf{u}^\top \mathbf{v}_2$, then v_1 is similar to v_2 in terms of user u). Since the recommendation models are able to measure the similarity, models are capable of grouping items based on the corresponding similarity measurement. Suppose there are n items and d groups ($n \gg d$), we denote the set of items

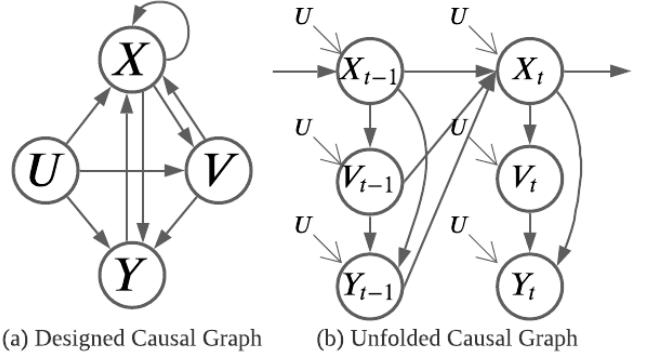


Figure 1: In the causal graph, U is user, V is the exposed item, X is user interaction history, and Y is preference score. (a) is our designed causal graph, more details are introduced in Section 3.1. (b) is the temporal unfolded causal graph, which guides the design of the proposed framework. We reorganize causal graph using U as exogenous variable to make it clearer. More details are introduced in Section 4.

as $\mathcal{I} = \{i_1, i_2, \dots, i_n\}$ and the set of groups as $\mathcal{C} = \{c_1, c_2, \dots, c_d\}$. We denote the grouping function as $A : \mathcal{I} \rightarrow \mathcal{C}$, which returns a group for each item based on similarity.

If the system does not consider echo chambers carefully, then similarity-driven recommendation tends to recommend items that are similar to previously interacted items, i.e., they may belong to the same group as the items in the user's interaction history [23, 27, 33]. This tendency stems from the principle of CF-based models that users have similar tastes in the past are likely to have similar interests in the future [10]. For example, a simple MF model such as [24] recommends items with higher preference which is calculated as $\mathbf{u}^\top \mathbf{v}$. A well-trained MF model produces high preference scores on items within the interaction history, thus the recommended items with high preference scores would be similar to one or more items within the interaction history since they are both similar to the user embedding \mathbf{u} . Since the grouping is based on the similarity learned by the model, a recommended item will belong to the same group with its similar items in the interaction history.

We denote the capacity of items in the interaction history as m (echo chambers occur in the repetitive behaviors [15] thus the cold-start scenarios are not within the scope of our discussion). It is worth clarifying that m can be infinity, but in practice, the items interacted a long time ago may have little influence on current predictions. We represent the history as a sequence of items $\{h_1, h_2, \dots, h_m\}$, where $h_i \in \mathcal{I}$ (if the number of valid items \tilde{m} in the history is less than m , then the first $m - \tilde{m}$ items would be empty). When a new item is interacted and added to the user history, the first item would be removed from the history to keep the history at length m . Applying the grouping function A to the item sequence, we obtain a historical group sequence $\{A(h_1), A(h_2), \dots, A(h_m)\}$, $A(h_i) \in \mathcal{C}$, which can be simplified as $\{A_1, A_2, \dots, A_m\}$, $A_i \in \mathcal{C}$. Since there are d groups in total, there are d^m possible historical group sequences. When the user interacts with a new item, both the item sequence and the historical group sequence will be updated. Since the recommendation models that do not consider echo chambers

tend to recommend items that are similar with historical items (i.e., most likely belong to the same group), the group of the new item only depends on the current historical group sequence. Therefore, we can formulate this iterative process as a Markov chain.

We consider the states in the Markov chain as historical group sequences, therefore, there are d^m different states. The transition probabilities are determined by the policy of interacting with new items. To analyze the occurrence of echo chambers for different users, we categorize user behavior into the following three types.

- (1) Users only interact with recommended items.
- (2) Users completely ignore the recommendations.
- (3) Users interact with both recommended and not recommended items.

We will separately analyze the above three categories from the perspective of echo chambers.

3.2.2 Type 1. When users only interact with recommended items, the transition probability is fully determined by the recommendation model. As we mentioned before, recommendation models that do not consider echo chambers tend to recommend items belonging to the same group as historical items. We can define a general one-step transition probability as follows:

$$P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) = \begin{cases} [A'_m \in \{A_1, \dots, A_m\}] P_{A'_m} & \text{if } A'_j = A_{j+1}, j = 1, \dots, m-1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $[P]$ represents the Iverson bracket, which maps a statement into a binary value (i.e., takes value 1 if the statement P is True and 0 otherwise) and $P_{A'_m}$ is the probability that the recommended items belong to group A'_m . Based on the one-step transition probability defined as Eq.(1), we are able to discover some special states. If $A_1 = A_2 = \dots = A_m$, we have

$$P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) = \begin{cases} 1 & \text{if } A'_1 = A'_2 = \dots = A'_m = A_i \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

We call these special states *absorbing states*; once users enter these states, they can no longer exit.

DEFINITION 1. [49] (*Absorbing Markov Chains*) A Markov chain is an absorbing Markov chain if:

- (a) there is at least one absorbing state and
- (b) it is possible to go from any state to at least one absorbing state in a finite number of steps.

Based on the transition probabilities shown in Eq.(2), there are d absorbing states in total. Meanwhile, based on Eq.(1), we can derive that $P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) > 0$ if $A'_m = A_m, A'_j = A_{j+1}, j = 1, \dots, m-1$. Therefore, it is possible to go from any states to an absorbing state in m steps. According to Definition 1, we can claim that in this setting, we have an absorbing Markov chain.

As an absorbing Markov chain with d absorbing states, the transition matrix P can be transformed into a block matrix, producing the following k -step transition matrix when $k \rightarrow \infty$ [49]:

$$P = \begin{bmatrix} Q & R \\ \mathbf{0} & I_d \end{bmatrix} \Rightarrow \lim_{k \rightarrow \infty} P^k = \begin{bmatrix} \mathbf{0} & (I - Q)^{-1}R \\ \mathbf{0} & I_d \end{bmatrix} \quad (3)$$

In summary, if a user keeps interacting with the recommended items, the interacted items will eventually belong to the same group and will not include other groups any more. This is how echo chambers lead to homogeneity and polarization.

3.2.3 Type 2. If users completely ignore the recommendation model, then each group has a nonzero probability to be interacted. We can define the one-step transition probability as follows:

$$P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) = \begin{cases} P'_{A'_m} & \text{if } A'_j = A_{j+1}, j = 1, \dots, m-1 \\ 0 & \text{else} \end{cases} \quad (4)$$

where $P'_{A'_m}$ is the probability that user chooses to interact with the item in the A'_m group without the influence of recommendation.

Based on the one-step transition probability shown in Eq.(4), we can calculate the k -step transition matrix. Since $P'_{A'_m}$ is a nonzero value for any group, there is no absorbing state, which means that the interaction history will not fall into the same group and remain unchanged. We take uniform distribution as a simple example (i.e., $P'_{A'_m} = 1/d$), in this case, we can obtain the following k -step transition matrix if k is large enough:

$$P^k = \left[\frac{1}{d^m} \right]_{d^m \times d^m}, \text{ if } k \geq \log m + 1 \quad (5)$$

For users in this type, users will avoid the drawbacks brought by echo chambers as long as each group has a nonzero probability to be interacted. However, considering the extremely large scale of items in practice, completely ignoring the recommendations will lose the benefits offered by personalized recommender systems.

3.2.4 Type 3. For the last type of users, we denote probability $p \in (0, 1)$ as following recommendation models and probability $1-p$ as not following recommendation models. We can then define the one-step transition probability as follows:

$$P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) = \begin{cases} P_{A'_m}^f p + P'_{A'_m} (1-p) & \text{if } A'_j = A_{j+1}, j = 1, \dots, m-1 \\ 0 & \text{else} \end{cases} \quad (6)$$

where $P_{A'_m}^f = [A'_m \in \{A_1, \dots, A_m\}] P_{A'_m}$ represents the probability of the recommended items belonging to group A'_m as Eq.(1), and $P'_{A'_m}$ is the probability that user chooses to interact with the item in the A'_m group without the influence of recommendation.

Notice that the Markov chain is irreducible. In other words, it is possible to result in any state from an arbitrary starting state after m steps, because $P(\{A'_1, \dots, A'_m\} | \{A_1, \dots, A_m\}) > 0, \forall A'_m \in C, A'_j = A_{j+1}, j = 1, \dots, m-1$. Since every state is accessible from any other state, In Type 3, the effect of echo chambers could be alleviated to some extent. When p is close to 1, the effect of echo chamber would be increased. When p is close to 0, the effect of echo chamber would be mitigated, but it also loses the benefits of personalized recommendation. In other words, if probability p is properly chosen, users will not only benefit from the recommendation results but also avoid homogeneity brought by echo chambers.

Given the evidence of echo chambers in some existing works [7, 15], the above analysis based on our designed causal graph confirms the rationality of the causal graph. Based on our analysis, if

Symbol	Definition
t	The timestamp
U, u	User variable and corresponding specific values
V_t, v_t	The variable denotes exposed item at time t and the corresponding specific values
X_t, x_t	The variable denotes user history at time t and the corresponding specific values
Y_t, y_t	The variable denotes the preference score at time t and the corresponding specific values
x_t^*	The observed user history at time t in real world
x_t'	The counterfactual user history at time t which is unobserved in real world
v_t^*	The observed exposed item at time t in real world
v_t'	The counterfactual exposed item at time t which is unobserved in real world
y_t^*	The observed preference score at time t in real world
y_t'	The counterfactual preference score at time t which is unobserved in real world

Table 1: Notations

the recommender system does not consider echo chambers, whether recommendation will lead to echo chamber is determined by user behavior. If the user actively explores the items of interest instead of only passively interacting with recommended items, the user may not be affected by echo chambers. If the user only passively interacting with the recommended items and the recommendation models do not consider echo chambers, then the user is likely to be affected by echo chambers.

In real-world applications, we cannot control the user behaviors. Therefore, mitigating echo chambers from the perspective of model design will play a significant role for all kinds of users. The most straightforward way to mitigate echo chamber is to use a random recommendation policy. However, this is against the ultimate goal of recommendation, which is to accurately capture user preference and recommend items that user may like. As a result, it is impractical to give up the recommendation accuracy for the pursuit of mitigating echo chambers. Therefore, it is essential to mitigate echo chambers while maintaining the recommendation performance. In the next section, we will introduce our proposed framework for mitigating echo chambers.

4 METHODOLOGY

In this section, we will introduce our Dynamic Causal Collaborative Filtering (∂ CCF) framework in detail. The primary notations used throughout this section are detailed in Table 1.

4.1 Preference Score Estimation

Our causal graph in Figure 1, with corresponding explanations in Section 3.1, describes a dynamic data generation process in recommendation. Therefore, the proposed causal graph with loops can be temporally unfolded. We capture a clip at time t and time $t-1$ shown in Figure 1(b). The unfolded causal graph represents the same data generation process as Figure 1(a) and it helps to mitigate echo chambers in each step.

In this paper, we establish a counterfactual query similar to existing causal recommendation models [51, 52, 54], i.e., what would

have happened if an item had been recommended, which can be mathematically represented as $P(y|u, do(v))$ [55]. In our case, we temporally unfold the causal graph and estimate the causal effect at each timestamp. Specifically, the desired estimation would be $P(y_t|u, do(v_t))$ at time t . When we unfold the designed causal graph and capture a snapshot at time t as in Figure 1(b), we can apply the back-door adjustment to estimate the post-intervention effect based on the observational data. Since the variable set $\{U, X_t\}$ satisfies the back-door criterion for estimating $P(y_t|u, do(v_t))$ (i.e., variable U blocks the back-door path $Y_t \leftarrow U \rightarrow V_t$ and variable X_t blocks the back-door path $Y_t \leftarrow X_t \rightarrow V_t$), we are able to calculate $P(y_t|u, do(v_t))$ as follows.

$$P(y_t|u, do(v_t)) = \sum_{x_t} P(y_t|u, v_t, x_t)P(x_t|u) \quad (7)$$

From Eq.(7), the key difference between our causal model and the traditional associative models is the existence of conditional probability $P(x_t|u)$. In the real world, we denote the conditional probability $P(x_t^*|u) = 1$ for an observed user history x_t^* at time t and $P(x_t'|u) = 0$ for an unobserved user history x_t' at time t . However, observing user history x_t^* does not imply that the user is destined to interact with items in x_t^* at time t . Considering the counterfactual world, if a user had a chance to be recommended different items, they may also interact with those different items, thus $P(x_t'|u)$ is not necessarily to be zero. As a result, calculating Eq.(7) requires counterfactual reasoning beyond the observational data in the real world [55]. In the following, we will introduce how we can leverage counterfactual reasoning to estimate the preference score.

4.2 Counterfactual Reasoning

In our framework, the purpose of counterfactual reasoning is not only to enable the calculation of Eq.(7) but also to break the feedback loop in Figure 1 to mitigate echo chambers. As we mentioned before, the user may have a chance to interact with different items that belong to the unobserved history x_t' at time t . Considering a record (u, v_t, y_t) at time t in the observational data, the user preference estimation $y_t = f_t(u, v)$ can be expressed in the following equations according to Eq.(7).

$$\begin{aligned} y_t &= f_t(u, v) \propto P(y_t|u, do(v_t)) \\ &= \sum_{\tilde{x}_t} P(y_t|u, v_t, \tilde{x}_t)P(\tilde{x}_t|u) = E_{\tilde{x}_t|u}[P(y_t|u, v_t, \tilde{x}_t)] \end{aligned} \quad (8)$$

Here we use \tilde{x}_t to represent possible user histories, including the factual history x_t^* and counterfactual histories x_t' . Based on Eq.(8), the estimation for $P(y_t|u, do(v_t))$ is the expected estimation of $P(y_t|u, v_t, \tilde{x}_t)$.

4.2.1 Generate Counterfactual Histories. We will use the generated counterfactual histories to calculate Eq.(8) and mitigate echo chambers. In particular, we design a heuristic-based approach for counterfactual history generation.

First, we take a look at an example of how the historical sequence is generated in the feedback loop. Consider an observed status (u, x_{t-1}^*) for user u at time $t-1$. Based on the observed history x_{t-1}^* , the system may recommend item v_{t-1}^* . If the user u passively accepts the recommended item v_{t-1}^* and likes it, then v_{t-1}^* and $y_{t-1}^* = 1$ would be a part of history x_t^* at time t . Based on our

analysis in Section 3.2, such users will be affected by echo chambers if the recommendation model does not consider echo chamber. We cannot force users to actively explore different items in real-world recommender systems. Instead, we let users actively explore different items in the counterfactual world. Following the principle of minimal changes in counterfactual reasoning [47, 48, 52, 55], we replace the observed interaction at time $t - 1$ to generate the counterfactual histories at time t :

$$x'_t \leftarrow x_{t-1}^*, v'_{t-1}, y'_{t-1} \quad (9)$$

Intuitively, this represents two possible situations: 1) the user still interacts with the recommended item v_{t-1}^* but has a different preference y'_{t-1} (i.e., $y'_{t-1} = 0$ if $y_{t-1}^* = 1$); 2) the user ignores item v_{t-1}^* and interacts with a different item v'_{t-1} and likes it (i.e., $y'_{t-1} = 1$). To mitigate echo chambers, the user should interact with an item which is not similar with the recommended item v_{t-1}^* . We select counterfactual item v'_{t-1} with least similarity to the recommended item v_{t-1}^* to maximally mitigate echo chambers.

$$v'_{t-1} = \arg \min_v \text{Similarity}(v, v_{t-1}^*) \quad (10)$$

Here the similarity is calculated by the dot product between the embedding of the two items. It is worth mentioning that the counterfactual item v'_{t-1} is not a real interaction nor an item to be recommended in reality.

4.2.2 Calculate the Expectation. Assume that we have generated n counterfactual histories $\{x_t^{(1)}, \dots, x_t^{(n)}\}$ for user u at time t . Then, we can calculate $P(y_t|u, do(v_t))$ based on the factual history x_t^* and the counterfactual histories $\{x_t^{(i)}\}_{i=1}^n$ according to Eq.(8). For simplicity, we consider $P(\tilde{x}_t|u)$ as a piece-wise uniform distribution over the factual and counterfactual histories, i.e.,

$$P(\tilde{x}_t|u) = \begin{cases} \alpha, & \text{when } \tilde{x}_t = x_t^* \\ \beta, & \text{when } \tilde{x}_t = x_t^{(i)}, i \in \{1, 2 \dots n\} \end{cases}, \alpha + n\beta = 1 \quad (11)$$

where α is the probability of the factual example x_t^* , and β is the probability of each counterfactual example $x_t^{(i)}$. Since x_t^* is already observed, we apply a higher probability to x_t^* than $x_t^{(i)}$, i.e., $\alpha > \beta > 0$. Then we have:

$$\begin{aligned} P(y_t|u, do(v_t)) &= \sum_{\tilde{x}_t} P(y_t|u, v_t, \tilde{x}_t) P(\tilde{x}_t|u) \\ &= \alpha P_g(y_t|u, v_t, x_t^*) + \beta \sum_{i=1}^n P_g(y_t|u, v_t, x_t^{(i)}) \end{aligned} \quad (12)$$

We use P_g to represent the probability estimation of the base recommendation algorithm.

As we mentioned in Section 4.2.1, the counterfactual histories intuitively represent the possible historical records if a user ignores the recommended items. Additionally, if we treat the factual history as a result of following recommender systems, the probability α in Eq.(12) can be considered as sharing the same meaning with probability p in Eq.(6). As we mentioned in Section 3.2, if the probability p is properly chosen, users will benefit from personalized recommendation while avoiding homogeneity brought by echo chambers. We will explore the effect of probability α in Section 5.7.

Table 2: The Statistic of the Datasets

Dataset	Phase	# Users	# Items	# Interactions
Movielens-1m	All	6,040	3,706	1,000,209
	Phase1	1,611	3,322	226,543
	Phase2	2,878	3,505	382,025
	Phase3	2,738	3,627	391,641
Electronics	All	33,602	16,448	788,143
	Phase1	8,495	3,679	39,489
	Phase2	29,798	13,820	377,460
	Phase3	31,193	15,162	371,194

5 EXPERIMENTS

In this section, we will first describe the evaluation metrics, datasets, baselines and implementation details and then provide our results and discussion.

5.1 Evaluation of Echo Chamber Effects

Echo chambers will result in homogeneity and polarization, hence over time the system will narrow the user exposure towards some specific content. Therefore, similar to [15], we evaluate the effect of echo chambers by measuring the changes of content diversity.

According to [15, 23], we use the average of the pairwise distance (i.e., Euclidean distance between item embeddings) to measure the content diversity, and use the temporal changes of content diversity to measure the echo chamber effect of the recommender systems.

5.2 Dataset Description

Our experiments are conducted on two real-world datasets from *Amazon*¹ [35] and *Movielens*² [18]. More specifically, we use *Electronics* category from Amazon and *Movielens-1m* from Movielens.

In order to measure the effect of echo chambers by the temporal change of content diversity, we use two timestamps to chronologically split the data into three phases. The *Movielens-1m* dataset contains the data in 2000, so we split the data at July 31th, 2000 and November 20th, 2000. The *Electronics* dataset contains data spanning May 1996 to October 2018. We use December 31th, 2010 and June 30th, 2015 as two timestamps. The statistics of the datasets are summarized in Table 2.

For all datasets, following prior works, we consider ratings ≥ 4 as positive feedback (likes) and ratings ≤ 3 as negative feedback (dislikes). Meanwhile, for all three phases in two datasets, we apply leave-one-out to split training, validation and test data [63].

5.3 Baseline Models

We also employ other six frameworks for comparison, including one re-ranking framework and five causal learning frameworks. **MMR** [6] is a re-ranking model for information retrieval which tries to maximize the relevance and novelty in finally retrieved top-ranked items. **IPS** [44] is an Inverse Propensity Scoring-based model, which uses a propensity estimator to re-weight the training samples to eliminate popularity bias. More specifically, we apply the clipped propensity score as [40] to reduce the variance of IPS. **CAUSE** [5] creates two sets of embeddings for unbiased and biased

¹<https://nijianmo.github.io/amazon/>

²<https://grouplens.org/datasets/movielens/>

Methods	Movielens 1m												Electronics															
	Phase2						Phase3						Echo Chambers		Phase2						Phase3						Echo Chambers	
	n@10	H@10	un@10	uH@10	n@10	H@10	un@10	uH@10	Δ12	Δ13	n@10	H@10	un@10	uH@10	n@10	H@10	un@10	uH@10	Δ12	Δ13	n@10	H@10	un@10	uH@10	Δ12	Δ13		
GRU4Rec	Original	0.5293	0.7797	0.1939	0.3073	0.5072	0.7482	0.1848	0.2929	0.996	1.174	0.2726	0.4567	0.0816	0.1294	0.3206	0.5400	0.0942	0.1494	6.270	7.205							
	MMR	0.5267	0.7747	0.1925	0.3052	0.5102	0.7467	0.1900	0.3012	0.971	1.179	0.2614	0.4374	0.0820	0.1274	0.3156	0.5304	0.0939	0.1495	6.138	7.178							
	IPS	0.5233	0.7640	0.1919	0.3041	0.5001	0.7452	0.1786	0.2831	0.956	1.079	0.2678	0.4582	0.0788	0.1290	0.3183	0.5376	0.0909	0.1483	6.783	7.275							
	CausE	0.5310	0.7836	0.1954	0.3107	0.5058	0.7546	0.1837	0.2952	1.244	1.352	0.2764	0.4617	0.0829	0.1315	0.3123	0.5339	0.0891	0.1412	6.395	7.154							
	CCF	0.5327	0.7807	0.1972	0.3098	0.5091	0.7527	0.1865	0.2937	1.262	1.342	0.2838	0.4708	0.0846	0.1367	0.3276	0.5427	0.0952	0.1490	6.832	7.421							
	DICE	0.5372	0.7882	0.1986	0.3148	0.5026	0.7416	0.1822	0.2887	1.244	1.349	0.2862	0.4727	0.0885	0.1402	0.3178	0.5320	0.0943	0.1495	7.052	7.630							
	MACR	0.5360	0.7815	0.2009	0.3184	0.5067	0.7441	0.1834	0.2906	1.167	1.255	0.2699	0.4511	0.0808	0.1280	0.3200	0.5377	0.0947	0.1500	6.163	7.218							
STAMP	Original	0.5002	0.7401	0.1813	0.2873	0.4776	0.7214	0.1681	0.2665	2.120	2.244	0.2839	0.4687	0.0878	0.1391	0.3065	0.5206	0.0882	0.1398	3.910	4.152							
	MMR	0.4845	0.7216	0.1784	0.2827	0.4667	0.6962	0.1705	0.2703	2.118	2.240	0.2788	0.4594	0.0883	0.1399	0.3025	0.5138	0.0873	0.1383	3.923	4.133							
	IPS	0.4920	0.7401	0.1743	0.2763	0.4761	0.7199	0.1679	0.2661	1.779	1.882	0.2908	0.4764	0.0903	0.1431	0.3258	0.5483	0.0952	0.1508	3.933	4.318							
	CausE	0.4999	0.7483	0.1847	0.2885	0.4816	0.7267	0.1667	0.2705	2.116	2.231	0.2828	0.4754	0.0878	0.1458	0.3172	0.5402	0.0928	0.1411	3.934	4.237							
	CCF	0.5037	0.7412	0.1802	0.2867	0.4815	0.7260	0.1683	0.2691	2.112	2.249	0.2926	0.4795	0.0901	0.1428	0.3247	0.5440	0.0963	0.1527	3.933	4.254							
	DICE	0.5080	0.7612	0.1879	0.3011	0.4868	0.7295	0.1742	0.2801	2.187	2.295	0.2818	0.4618	0.0865	0.1343	0.2909	0.5024	0.0802	0.1272	3.918	4.205							
	MACR	0.5023	0.7462	0.1858	0.2945	0.4746	0.7158	0.1710	0.2710	2.082	2.203	0.2912	0.4766	0.0910	0.1442	0.3082	0.5239	0.0881	0.1396	3.903	4.177							
NCR	Original	0.4853	0.7579	0.1597	0.2531	0.4725	0.7331	0.1572	0.2492	1.028	1.074	0.2522	0.4287	0.0722	0.1145	0.2737	0.4739	0.0750	0.1189	4.705	5.002							
	MMR	0.4853	0.7579	0.1633	0.2588	0.4560	0.7218	0.1470	0.2329	0.953	1.027	0.2549	0.4363	0.0723	0.1145	0.2720	0.4734	0.0749	0.1187	4.676	4.918							
	IPS	0.4833	0.7622	0.1561	0.2474	0.4626	0.7410	0.1427	0.2262	1.003	1.077	0.2596	0.4345	0.0773	0.1237	0.2845	0.4887	0.0796	0.1261	4.600	4.920							
	CausE	0.4959	0.7676	0.1637	0.2613	0.4794	0.7369	0.1685	0.2533	1.059	1.105	0.2562	0.4327	0.0766	0.1217	0.2780	0.4859	0.0774	0.1241	4.595	4.937							
	CCF	0.4824	0.7533	0.1603	0.2497	0.4751	0.7403	0.1598	0.2478	1.015	1.087	0.2521	0.4306	0.0754	0.1227	0.2853	0.4929	0.0792	0.1255	4.486	4.993							
	DICE	0.4866	0.7526	0.1579	0.2501	0.4714	0.7275	0.1537	0.2473	1.032	1.050	0.2590	0.4346	0.0769	0.1235	0.2754	0.4791	0.0759	0.1174	4.650	4.925							
	MACR	0.4936	0.7615	0.1640	0.2599	0.4746	0.7233	0.1631	0.2586	1.043	1.060	0.2592	0.4342	0.0775	0.1228	0.2757	0.4798	0.0753	0.1193	4.488	4.965							
dCCF	Original	0.4941	0.7547	0.1590	0.2724	0.4659	0.7162	0.1603	0.2541	0.936	0.985	0.2614	0.4362	0.0784	0.1242	0.2781	0.4762	0.0787	0.1248	4.323	4.841							

Table 3: Overall performance of applying our framework on three recommendation models. The recommendation performance is evaluated as a ranking task. n, H, un, uH represent nDCG, Hit, u_nDCG and u_Hit respectively. The effect of echo chambers is measured by change of content diversity (the lower the better). We calculate the content diversity at each phase and report the change of content diversity. We use Δ_{ij} to denote the change of content diversity between phase i and phase j . The improved performance is bold and the best is underlined.

data separately and regularization is applied to force the two sets of embeddings similar. CCF [55] is a causal collaborative filtering framework which uses counterfactual histories and applies counterfactual constraints to estimate causal preference. DICE [64] is a framework for popularity bias problem, which disentangles user interest and conformity into two sets of embeddings. MACR [53] is a model-agnostic framework for alleviating popularity bias issue in recommender systems, which performs multi-task training according to causal graph to assess the contribution of different causes on the ranking score. Finally, we use ∂CCF to denote our dynamic causal collaborative filtering framework.

We apply all above frameworks on three recommendation models, including two sequential models and a reasoning model. **GRU4Rec** [19] is a session-based recommendation model, which uses recurrent neural networks—in particular, Gated Recurrent Units (GRU)—to capture sequential patterns. **STAMP** [37] is the Short-Term Attention/Memory Priority model, which takes attention mechanism to model both short-term and long-term user preferences. **NCR** [8] is the Neural Collaborative Reasoning model, which employs neural logic reasoning for recommendation in a logical space.

5.4 Implementation Details

For recommendation performance, we evaluate models as a top- K recommendation task. For each user in the validation and test set, we randomly sample 100 negative items for ranking evaluation, where the negative items are either negative feedback items (i.e., items that user dislikes) or non-interacted items. The models are evaluated on four metrics, two are nDCG@10 and Hit@10 metrics calculated by ranking on sampled testing data, and the other two are nDCG@10 and Hit@10 metrics calculated by corrected rank

as [25] to get an unbiased evaluation (denote as u_nDCG@10 and u_Hit@10). For performance of mitigating echo chambers, we evaluate the change of content diversity (i.e., introduced in Section 5.1) based on the list of recommendations with length 10.

For the hyper-parameters, we take the embedding dimension as 64, the structure of neural networks is a two-layer MLP with dimension 64. We set the learning rate to 0.001 and the ℓ_2 -regularization weight to 1e-6. For all base models, we set the maximum length of history as 10. For our framework, we generate 9 counterfactual histories for each positive user-item pair (i.e., n in Eq.(12)) and set the factual probability α as 0.3. We will discuss the influence of these two parameters in Section 5.7. We report the performance of each model with the best performance on validation data based on nDCG@10. For all base models, We first train the base model on phase 1 data to make sure the content diversity at phase 1 is consistent before and after applying the frameworks. Then, on phase 2 and phase 3 data, we separately train the base model with and without frameworks to measure the changes of content diversity.

5.5 Results and Discussions

In Table 3, we present the overall performance of applying our framework and baseline frameworks on three base recommendation models over two datasets, including recommendation performance (nDCG@10, Hit@10, u_nDCG@10 and u_Hit@10) and echo chambers evaluation (the changes of content diversity).

From Table 3, we can observe that the effects of echo chambers have been mitigated (i.e., the change of content diversity has been reduced) after applying our ∂CCF framework for all three base models over two datasets. Averaging the change of content diversity across all three base models over two phases of both datasets, the change

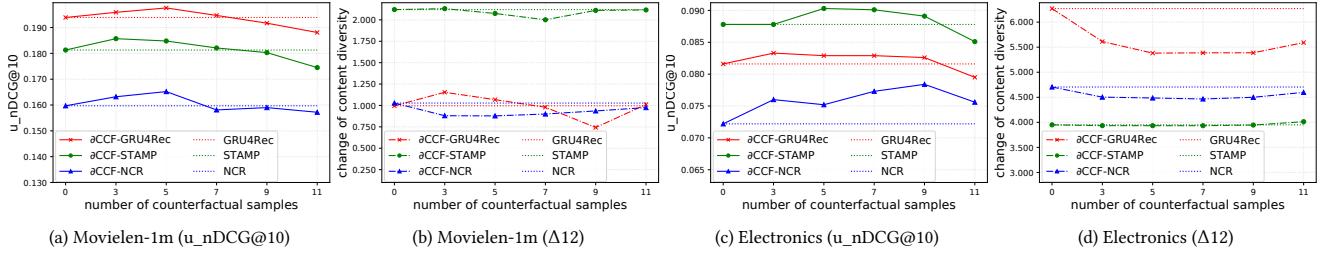


Figure 2: The influence of different number of counterfactual samples (n in Eq.(12)). (a) and (c) are recommendation performance ($u_{nDCG}@10$) on phase 2 of the two datasets. (b) and (d) are the change of content diversity ($\Delta 12$ in Table 3) on phase 2 of the two datasets. The change of content diversity is the lower the better.

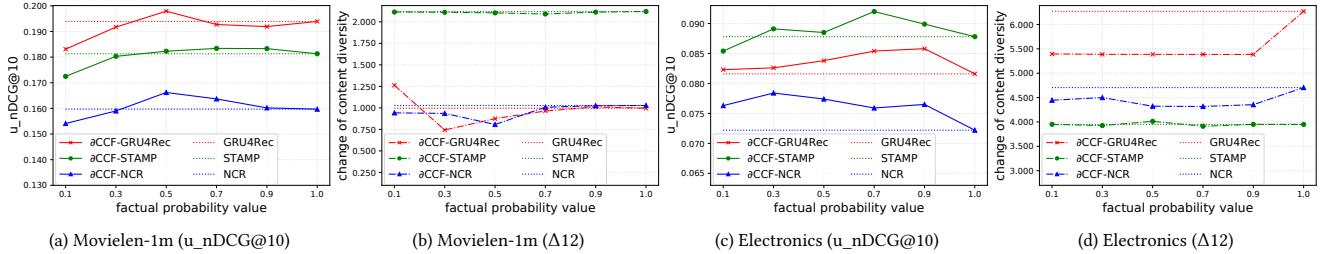


Figure 3: The influence of different factual probability (α in Eq.(12)). (a) and (c) are recommendation performance ($u_{nDCG}@10$) on phase 2 of the two datasets. (b) and (d) are the change of content diversity ($\Delta 12$ in Table 3) on phase 2 of the two datasets. The change of content diversity is the lower the better.

of content diversity has been reduced by 7.2%. The largest improvement 25.4% is obtained at phase 2 of the MovieLens-1m dataset after applying on GRU4Rec. For the recommendation performance, we can observe that applying our framework over the base models will not hurt the recommendation performance. We can achieve comparable recommendation performance after applying our framework. When averaging across all recommendation metrics on all datasets and base models, our ∂ CCF framework achieves 1% improvement on recommendation performance. The largest improvement is 8.6% on $u_{nDCG}@10$ of NCR at phase2 of the Electronics dataset. Therefore, our framework is capable of mitigating the echo chambers without sacrificing much recommendation performance.

The baseline frameworks include five causal learning frameworks (i.e., IPS, CausE, CCF, DICE, MACR) and one re-ranking framework (i.e., MMR). For the five causal learning frameworks, the goals of these causal learning frameworks are improving recommendation performance. Therefore, the best recommendation performance (i.e., the underlined values on recommendation metrics in Table 3) are usually obtained by those causal learning frameworks. Although these causal learning baseline frameworks can be helpful in mitigating echo chambers in some cases, the best performance on mitigating echo chambers is provided by our ∂ CCF framework in most cases. Among the 2 (datasets) \times 2 (phases) \times 3 (base models) = 12 cases, the ∂ CCF framework achieves 9 best performance on mitigating echo chambers. For the re-ranking framework MMR, it tries to maximize both the relevance and the novelty, which is an accuracy-diversity trade-off. According to Table 3, our ∂ CCF framework achieves better performance of mitigating echo chambers (i.e., achieves 9/12 best performance on mitigating echo chambers). Comparing the recommendation performance with the base model,

MMR decreases the recommendation performance by 1% when averaging across all recommendation metrics on all datasets and base models. Compared with MMR, ∂ CCF can incorporate counterfactual reasoning to avoid hurting the recommendation performance.

In summary, our framework gets better performance than other baseline frameworks on mitigating echo chambers without sacrificing the recommendation performance in most cases.

5.6 User Satisfaction

One consequence of echo chambers is gradually hurting user's satisfaction. Following [11], we evaluate models with the cumulative satisfaction to show the performance of mitigating echo chambers. Specifically, user satisfaction is defined as:

$$\text{Satisfaction} = \begin{cases} \text{Interest} & \text{If no echo chambers} \\ 0 & \text{Else} \end{cases} \quad (13)$$

It is required to obtain the interest of all possible user-item interactions, which cannot be satisfied by *MovieLens-1m* and *Electronics*. Therefore, we use *KuaiRec*³ dataest, which is a real-world dataset that contains a fully observed user-item interaction matrix (i.e., each user has feedback on each video). We define the interest of a user-item interaction as the video watching ratio which is the ratio of watching time length to the total video length. During the training, we consider watching ratio ≥ 2.0 as positive feedback (like) and watching ratio < 2.0 as negative feedback (dislike) as noted by authors. Following the exit mechanism designed by [11], the interaction process ends if the user feels bored. Specifically, for the most recent N recommended items, if at least N_q items in the

³<https://chongminggao.github.io/KuaiRec/>

	GRU4Rec		STAMP		NCR	
	CS	Len	CS	Len	CS	Len
Original	1.333	26	2.987	46	2.519	49
IPS	2.159	40	1.827	35	1.412	34
CausE	1.857	32	2.734	42	2.459	47
CCF	2.577	43	3.152	38	2.534	49
DICE	1.994	36	2.859	44	2.830	52
MACR	2.960	38	3.036	52	3.107	37
∂ CCF	3.597	54	3.219	58	2.998	57

Table 4: The average cumulative satisfaction and interaction length on KuaiRec dataset. CS represents cumulative satisfaction and Len represents interaction length. The best performance is highlighted.

N items share at least one attribute with the current recommended item, then the interaction process ends. Same as [11], we set N as 1 and N_q as 1, and apply softmax sampling to generate the recommendation, then add the recommended item to users' history for the next round recommendation. We report the average cumulative satisfactions and the interaction length in Table 4. The MMR framework is not included because the framework is a re-ranking framework, which has the same prediction results as the original model when recommending one item in each round.

From the results in Table 4, we can observe that the ∂ CCF framework achieves the best performance on cumulative satisfaction and interaction length in most cases. Without considering the echo chambers, the model may have higher satisfaction in a single round. However, such recommendation will narrow down to certain contents leading to echo chambers, which in turn makes users feel bored and quit. In this case, it may actually hurt users' satisfaction in the long run. Instead, the ∂ CCF framework may help users to explore different content to mitigate echo chambers, which may not achieve the best satisfaction in a single round but will obtain higher user satisfaction in the long run.

5.7 Sensitivity Analysis

In this section, we will discuss the influence of the two important parameters in our framework from the perspective of recommendation performance and mitigating echo chambers. One is the number of counterfactual histories (i.e., n in Eq.(12)). The other is the value of factual probability (i.e., α in Eq.(12)).

5.7.1 The number of counterfactual histories. Intuitively, in our model, each counterfactual history represents a possible result of the user actively exploring different items in the counterfactual world. We change the number of counterfactual histories while keeping other parameters fixed. The results are in Figure 2, including recommendation performance and the effect of echo chambers.

For the recommendation performance (i.e., (a) and (c) in Figure 2), we can observe that when the number of counterfactual histories is small (larger than 0), the recommendation performance is better than the base model in most cases, which means that proper counterfactual reasoning will make recommendation models better capture users' preference thus improve recommendation performance. However, when the number of counterfactual histories is too large, given that the factual probability is fixed, the probability

of each counterfactual history (i.e., β in Eq.(11)) is small. In this case, too many counterfactual histories may introduce too much noise into the recommendation models and thus hurt the recommendation performance. For the performance on mitigating echo chambers (i.e., (b) and (d) in Figure 2), we can observe that introducing counterfactual histories is helpful in mitigating echo chambers in most cases. However, when the number of counterfactual histories is too large, the items in counterfactual histories may not be well trained because the probability of each counterfactual history is too small, which will lead to worse performance on mitigating echo chambers. A proper number of counterfactual histories is required to mitigate the echo chambers without compromising the recommendation performance.

5.7.2 The value of factual probability. The value of factual probability determines the distribution over factual and counterfactual histories. A larger value of factual probability represents more weights over the factual history. We tune the value of factual probability α while keeping the other parameters fixed. We plot the recommendation performance as well as the effect of echo chambers in Figure 3. We can see that when the value is too small, the counterfactual histories will dominate the predictions, and thus may mislead the recommendation, but the framework still can mitigate the echo chambers in most cases. When the value is close to 1, the factual histories will dominate the predictions, thus the recommendation performance and the performance on mitigating echo chambers are closer to the base models.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we show that DAG causal graphs cannot fully describe the dynamic data-generation process. As a result, we design a directed cyclic causal graph to represent the dynamic nature of recommender systems. Besides, the dynamic process may result in unwanted effects such as echo chambers. To theoretically understand echo chambers, we represent user-system interaction as a Markov Process and group users into three categories based on users' behaviors. Mathematically, for each group of users, we analyze whether the recommendation will lead to an echo chamber. In addition to the above theoretical contributions, we also design a Dynamic Causal Collaborative Filtering (∂ CCF) framework, which takes the back-door adjustment to estimate user preferences and leverages counterfactual reasoning to mitigate echo chambers. Experiments on real-world datasets show that our framework can mitigate echo chambers while achieving comparable recommendation performance with the base models.

∂ CCF is a flexible framework for dynamic analysis of intelligent systems. In this work, we studied the echo chamber of recommender systems, while in the future, we can take ∂ CCF to explore other dynamics such as popularity shifting, influence analysis and emerging topics. We can also conduct dynamic analysis of other systems such as social networks, online forums and conversational AI.

Acknowledgment. This work was supported in part by NSF IIS 1910154, 2007907 and 2046457. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

REFERENCES

[1] David Paul Allen, Henry Jacob Wheeler-Mackta, and Jeremy R Campo. 2017. The effects of music recommendation engines on the filter bubble phenomenon. *Interactive Qualifying Projects* (2017).

[2] Cigdem Aslay, Antonis Matakos, Esther Galbrun, and Aristides Gionis. 2018. Maximizing the diversity of exposure in a social network. In *ICDM*. 863–868.

[3] Eytan Bakshy, Solomon Messing, and Lada A Adamic. 2015. Exposure to ideologically diverse news and opinion on Facebook. *Science* 348, 6239 (2015), 1130–1132.

[4] Sven Banisch and Eckehard Olbrich. 2019. Opinion polarization by learning from social feedback. *The Journal of Mathematical Sociology* 43, 2 (2019), 76–103.

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

[6] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*.

[7] Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. 2018. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *RecSys*. 224–232.

[8] Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. 2021. Neural Collaborative Reasoning. In *Proceedings of the Web Conference 2021*. 1516–1527.

[9] Uthsva Chitra and Christopher Musco. 2020. Analyzing the impact of filter bubbles on social network polarization. In *WSDM*. 115–123.

[10] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends in HCI* 4, 2 (2011), 81–173.

[11] Chongming Gao, Wenqiang Lei, Jiawei Chen, Shiqi Wang, Xiangnan He, Shijun Li, Biao Li, Yuan Zhang, and Peng Jiang. 2022. CIRS: Bursting Filter Bubbles by Counterfactual Interactive Recommender System. *arXiv:2204.01266* (2022).

[12] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. 2018. Political discourse on social media: Echo chambers, gate-keepers, and the price of bipartisanship. In *WWW*. 913–922.

[13] Yingqiang Ge, Shuchang Liu, Zuohui Fu, Juntao Tan, Zelong Li, Shuyuan Xu, Yunqi Li, Yikun Xian, and Yongfeng Zhang. 2022. A Survey on Trustworthy Recommender Systems. *arXiv:2207.12515* (2022).

[14] Y. Ge, J. Tan, Y. Zhu, Y. Xia, J. Luo, S. Liu, Z. Fu, S. Geng, Z. Li, and Y. Zhang. 2022. Explainable Fairness in Recommendation. In *SIGIR*.

[15] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding echo chambers in e-commerce recommender systems. In *SIGIR*. 2261–2270.

[16] Azin Ghazimatin, Oana Balalau, Rishiraj Saha Roy, and Gerhard Weikum. 2020. PRINCE: Provider-side interpretability with counterfactual explanations in recommender systems. In *WSDM*. 196–204.

[17] Madelyn Glymour, Judea Pearl, and Nicholas P Jewell. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.

[18] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *AcM transactions on interactive intelligent systems* 5, 4 (2015), 1–19.

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

[20] Martin Hilbert, Saifuddin Ahmed, Jaeho Cho, Billy Liu, and Jonathan Luu. 2018. Communicating with algorithms: A transfer entropy analysis of emotions-based escapes from online echo chambers. *Communication Methods & Measures* (2018).

[21] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. 2019. Degenerate feedback loops in recommender systems. In *AIES*. 383–390.

[22] Dimitris Kalimeris, Smriti Bhagat, Shankar Kalyanaraman, and Udi Weinsberg. 2021. Preference Amplification in Recommender Systems. In *KDD*. 805–815.

[23] Marius Kaminskas and Derek Bridge. 2016. Diversity, serendipity, novelty, and coverage: survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems* (2016).

[24] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[25] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *KDD*. 1748–1757.

[26] Tyll Krueger, Janusz Szwabiński, and Tomasz Weron. 2017. Conformity, anti-conformity and polarization of opinions: insights from a mathematical model of opinion dynamics. *Entropy* 19, 7 (2017), 371.

[27] Mátévž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems—A survey. *Knowledge-based systems* 123 (2017), 154–162.

[28] Yunqi Li, Hanxiong Chen, Juntao Tan, and Yongfeng Zhang. 2022. Causal factorization machine for robust recommendation. In *JCDL*. 1–9.

[29] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. 2022. Fairness in Recommendation: A Survey. *arXiv:2205.13619* (2022).

[30] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness Based on Causal Notion. In *SIGIR*.

[31] 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*. 831–840.

[32] Antonis Matakos and Aristides Gionis. 2018. Tell me Something My Friends do not Know: Diversity Maximization in Social Networks. In *ICDM*.

[33] Judith Möller, Damian Trilling, Natali Helberger, and Bram van Es. 2018. Do not blame it on the algorithm: an empirical assessment of multiple recommender systems and their impact on content diversity. *Info., Comm. & Society* (2018).

[34] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2014. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *WWW*. 677–686.

[35] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*.

[36] Nicola Perra and Luis EC Rocha. 2019. Modelling opinion dynamics in the age of algorithmic personalisation. *Scientific reports* 9, 1 (2019), 1–11.

[37] Qiao Liu, Yifei Zeng, Refuoe Mokhosi, Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *KDD*.

[38] Zhen Qin, Suming J Chen, Donald Metzler, Yongwoo Noh, Jingzheng Qin, and Xuanhui Wang. 2020. Attribute-based propensity for unbiased learning in recommender systems: Algorithm and case studies. In *KDD*. 2359–2367.

[39] Marten Risius, Okan Aydinguel, and Maximilian Haug. 2019. *Towards an understanding of conspiracy echo chambers on Facebook*. AISel.

[40] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased recommender learning from missing-not-at-random implicit feedback. In *WSDM*. 501–509.

[41] Masahiro Sato, Shin Kawai, and Hajime Nobuhara. 2019. Action-triggering recommenders: Uplift optimization and persuasive explanation. In *ICDM Workshops*.

[42] Masahiro Sato, Janmajay Singh, Sho Takemori, Takashi Sonoda, Qian Zhang, and Tomoko Ohkuma. 2019. Uplift-based evaluation and optimization of recommenders. In *RecSys*. 296–304.

[43] Masahiro Sato, Sho Takemori, Janmajay Singh, and Tomoko Ohkuma. 2020. Unbiased learning for the causal effect of recommendation. In *RecSys*. 378–387.

[44] T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *ICML*.

[45] Ana-Andreea Stoica and Augustin Chaintreau. 2019. Hegemony in Social Media and the effect of recommendations. In *Companion Proceedings of WWW*. 575–580.

[46] Cass R Sunstein. 2009. *Going to extremes: How like minds unite and divide*. Oxford University Press.

[47] Juntao Tan, Shijie Geng, Zuohui Fu, Yingqiang Ge, Shuyuan Xu, Yunqi Li, and Yongfeng Zhang. 2022. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In *WWW*. 1018–1027.

[48] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual explainable recommendation. In *CIKM*.

[49] Anders Tolver. 2016. An introduction to Markov chains. *Department of Mathematical Sciences, University of Copenhagen* (2016).

[50] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *SIGIR (SIGIR '21)*. 1627–1631.

[51] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. 2020. Causal inference for recommender systems. In *RecSys*. 426–431.

[52] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Counterfactual data-augmented sequential recommendation. In *SIGIR*. 347–356.

[53] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *KDD*. 1791–1800.

[54] Kun Xiong, Wenwen Ye, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, Binbin Hu, Zhiqiang Zhang, and Jun Zhou. 2021. Counterfactual Review-based Recommendation. In *CIKM*.

[55] Shuyuan Xu, Yingqiang Ge, Yunqi Li, Zuohui Fu, Xu Chen, and Yongfeng Zhang. 2021. Causal Collaborative Filtering. *arXiv:2102.01868* (2021).

[56] Shuyuan Xu, Juntao Tan, Shelby Heinecke, Jia Li, and Yongfeng Zhang. 2021. Deconfounded Causal Collaborative Filtering. *arXiv:2110.07122* (2021).

[57] Longqi Yang, Yin Cui, Yuan Xuan, Chenyang Wang, Serge Belongie, and Deborah Estrin. 2018. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *RecSys*. 279–287.

[58] Mengye Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. Top-N Recommendation with Counterfactual User Preference Simulation. In *CIKM*.

[59] Junzhe Zhang and Elias Bareinboim. 2018. Fairness in decision-making—the causal explanation formula. In *AAAI*.

[60] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *SIGIR*. 367–377.

[61] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *Comput. Surveys* (2019).

[62] Yongfeng Zhang and Xu Chen. 2020. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020).

[63] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM*. 2329–2332.

[64] Yu Zheng, Chen Gao, Xiang Li, Xiangnan He, Yong Li, and Depeng Jin. 2021. Disentangling User Interest and Conformity for Recommendation with Causal Embedding. In *Proceedings of the Web Conference 2021*. 2980–2991.