# Causal Structure Representation Learning of Unobserved Confounders in Latent Space for Recommendation

HANGTONG XU, MIC Lab, College of Computer Science and Technology, Jilin University, China
YUANBO XU*, MIC Lab, College of Computer Science and Technology, Jilin University, China
CHAOZHUO LI, Key Laboratory of Trustworthy Distributed Computing and Service (MoE), Beijing University of Posts and Telecommunications, China
FUZHEN ZHUANG, Institute of Artificial Intelligence, Beihang University and Zhongguancun Laboratory, China

Inferring user preferences from users' historical feedback is a valuable problem in recommender systems. Conventional approaches often rely on the assumption that user preferences in the feedback data are equivalent to the real user preferences without additional noise, which simplifies the problem modeling. However, there are various confounders during user-item interactions, such as weather and even the recommendation system itself. Therefore, neglecting the influence of confounders will result in inaccurate user preferences and suboptimal performance of the model. Furthermore, the unobservability of confounders poses a challenge in further addressing the problem. Along these lines, we refine the problem and propose a more rational solution to mitigate the influence of unobserved confounders. Specifically, we consider the influence of unobserved confounders, disentangle them from user preferences in the latent space, and employ causal graphs to model their interdependencies without specific labels. By ingeniously combining local and global causal graphs, we capture the user-specific effects of confounders on user preferences. Finally, we propose our model based on Variational Autoencoders, named **C**ausal **S**tructure **A**ware **V**ariational **A**uto**e**ncoders (CSA-VAE) and theoretically demonstrate the identifiability of the obtained causal graph. We conducted extensive experiments on one synthetic dataset and nine real-world datasets with different scales, including three unbiased datasets and six normal datasets, where the average performance boost against several state-of-the-art baselines achieves up to 9.55%, demonstrating the superiority of our model. Furthermore, users can control their recommendation list by manipulating the learned causal representations of confounders, generating potentially more diverse recommendation results. Our code is available at Code-link[1].

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

Additional Key Words and Phrases: Causal structure, Preference modeling, Confounders, Variational inference

## 1 INTRODUCTION

Recommender systems play a vital role in information technology, aiming to assist users in discovering content or products that might align with their interests [32, 33]. User historical feedback data is a crucial basis for model

*Corresponding Author: Yuanbo Xu
[1] https://github.com/MICLab-Rec/CSA

Authors' Contact Information: Hangtong Xu, MIC Lab, College of Computer Science and Technology, Jilin University, Changchun, China, xuht21@mails.jlu.edu.cn; Yuanbo Xu, MIC Lab, College of Computer Science and Technology, Jilin University, Changchun, China, yuanbox@jlu.edu.cn; Chaozhuo Li, Key Laboratory of Trustworthy Distributed Computing and Service (MoE), Beijing University of Posts and Telecommunications, China, lichaozhuo@bupt.edu.cn; Fuzhen Zhuang, Institute of Artificial Intelligence, Beihang University and Zhongguancun Laboratory, Beijing, China, zhuangfuzhen@buaa.edu.cn.

Manuscript submitted to ACM

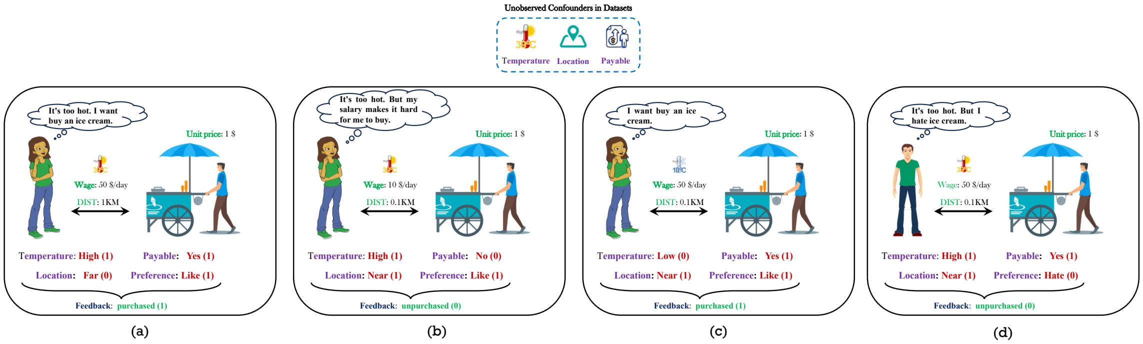


Fig. 1. An example illustrating that user preferences in the feedback data are influenced by both the users themselves and external confounders(e.g., temperature, location).

prediction of user preferences. Based on this, many highly effective methods have been proposed, such as MultiVAE [10], MacridVAE [13], etc. Existing work often assumes that user preferences in historical feedback data are noise-free, as shown in Figure 1 (a) and (d), the alignment of user preferences (like or hate) with the collected feedback data (purchased or unpurchased) is crucial, thus focusing on fitting the feedback data can achieve acceptable model performance.

However, various confounders inevitably influence users during the interaction process, affecting their final decisions, and the extent of the impact differs among users [4, 12]. As shown in Figure 1, user preferences are influenced by both the intrinsic characteristics of users (user-specific preference) and external confounders in the interaction environment (e.g., temperature, location). For example, when analyzing user feedback for a specific product like ice cream on an online retail platform, we observe varying user preferences due to confounders. Some users may exhibit opposite feedback regarding their preferences for ice cream. For instance, the wage at the time of purchase is a confounder. As shown in Figure 1 (b), it's a tough decision for users to spend a tenth of their daily salary on ice cream. On the other hand, when user preference is strong enough, confounders have less impact on the final decision. As Figure 1 (c) shows, the user gives positive feedback for ice cream due to their favorite flavor on cold days.

Nevertheless, relying solely on confounders to determine user preferences is also unreasonable [28]. Figures 1 (a) and (d) show that two users with the same confounder conditions give opposite feedback. Thus, it is evident that user preferences, as reflected in the feedback data, are a combination of the intrinsic characteristics of users and external confounders like wage, representing the interplay of intrinsic and extrinsic influences. This also explains why models improve performance when additional information, such as time and location, is incorporated. Unfortunately, most confounders are unobservable, and we cannot obtain corresponding labels from users as additional information. Hence, we cannot explicitly model confounders to separate them from user preferences, and addressing the dynamic impact of confounders on users remains a challenge.

To address these challenges, we first reformulate the user preference prediction problem by introducing the influence of confounders. In this way, user preferences in the feedback data stem from the combined impact of the inherent preferences of users and external confounders. We proposed a mild assumption of confounder independence to disentangle confounders from user preferences in the latent space. Specifically, we assumed that the influence of all confounders on each user originates from the same set of confounders, which ensures user independence of confounders. Furthermore, we found that confounders are not independent of each other. For example, the weather depends on users' location. We utilized a causal structural model (SEM) to represent the generation process of confounders. Specifically, we employed a binary matrix to denote the global causal graph, where directed edges signify the dependency relationships between confounders. However, for different users, the relationships between confounders also vary, sometimes opposite. Thus, we used an additional local causal graph to capture the user-specificity of confounders.

Furthermore, we demonstrate that the learned causal representations of confounders are controllable, potentially offering users fine-grained control over the objectives of their recommendation lists with the learned causal graphs. Finally, we combine the obtained causal representations of confounders with the inherent preferences of users to fit user preferences in historical feedback data, proposed a model based on Variational Autoencoder (VAE) named **C**ausal **S**tructure **A**ware **V**ariational **A**uto**e**ncoders (CSA-VAE) to learn causal representations of user preferences and confounders simultaneously. In addition, we theoretically proved the identifiability of the model.

The contributions of our work can be summarized as follows:

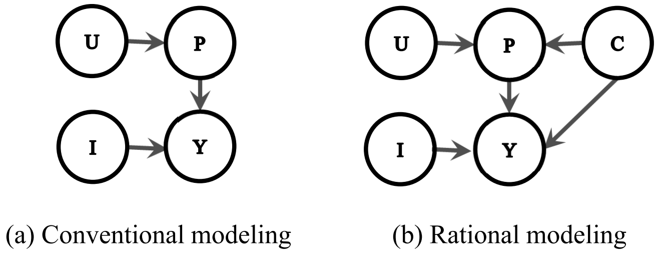
- We have re-formalized the problem of user preference prediction, providing a more reasonable modeling approach for user preferences.
- We introduced a mild assumption that allows for the independent representation of user preferences and the influence of confounders and utilized causal graphs to capture the dependencies among confounders. Furthermore, we provide proof of the identifiability of the causal graph and the visualization of learned confounder representations.
- We employed global and local causal graphs to capture the invariance and specificity between users and confounders. We proposed a model based on the Variational Autoencoder (VAE) to simultaneously learn causal representations of user preferences and confounders.
- We conducted extensive experiments on a synthetic dataset and nine real-world datasets with different scales, including three unbiased datasets and six normal datasets, where the average performance boost against several state-of-the-art baselines achieves up to 9.55%, demonstrating our model's effectiveness.
- We formalize the user-controlled recommendation task by integrating both latent preferences and confounders while incorporating causal interventions to give the user control over their preferences within the recommendation system.

Our code is publicly available at: https://github.com/MICLab-Rec/CSA.

## 2 RELATED WORK

### 2.1 Deconfound in Recommendation

With the increasing popularity of causal inference as a method to mitigate bias in recommender systems [34], researchers are paying more attention to the challenges posed by confounding biases. Confounding bias is prevalent in recommender systems due to various confounders. While some studies have addressed specific confounding biases, such as item popularity [24, 29, 39], many unobservable confounders may also exist. The mainstream approaches can be broadly categorized into two types: (1) [31, 42] utilize additional signals as instrumental or proxy variables to mitigate confounding bias. (2) [30, 43] consider a multiple-treatment setting and infer surrogate confounders from user exposure, incorporating them into the preference prediction model. (3) SEM-MacridVAE [25], CaD-VAE [23] and PlanRec [38] utilize additional signals to learn the latent causal structure of confounders and make recommendation.



(a) Conventional modeling (b) Rational modeling

Fig. 2. Conventional modeling versus a more rational modeling approach. Left is the conventional modeling, and right is the rational modeling proposed in this work. **U**→User; **I**→Item; **C**→Confounders; **P**→Preference in feedback data; **Y**→User feedback. The exogenous variables of nodes (e.g., **C**) are not displayed in the graph.

However, they did not address the challenge of confounders in the absence of corresponding labels. CSA-VAE simultaneously learned user causal preferences and graphs without corresponding labels and employed global and local causal graphs to capture the invariance and specificity between users and confounders.

### 2.2 Causal Structure Learning

We refer to causal representations constructed by causal graphs as causal representations. Over the past few decades, discovering causal graphs from purely observational data has garnered significant attention. [41] proposed NOTEARs with a fully differentiable DAG constraint for causal structure learning, [20]show the identifiability of learned causal structure from interventional data. The community has raised interest in combining causality and disentangled representation, and [7] proposed a method called CausalGAN, which supports ”do-operation” on images, but it requires the causal graph given as a prior.

We draw on key ideas from causal structure learning to enhance the application of latent structure learning in recommendations. Additionally, we identify a key challenge in applying shared latent structure to recommendations: confounding factors affect users differently. To address this, we design a personalized structure learning framework and personalized recommendations.

## 3 METHODOLOGY

This section thoroughly introduces our proposed model with the necessary theoretical proofs.

### 3.1 A More Rational Architecture

Predicting user preferences from their historical interaction data is a common training paradigm in the domain of recommender systems and is generally based on the fundamental assumption that the user preferences contained in the feedback data reflect the true preferences of the user. Based on such assumptions, the user preference prediction problem can be formulated as follows:

Problem 1. *Given the historical interaction data*[2] $x_u = \{x_{u,1}, ..., x_{u,n}\}$*, we can predict the user preference* $z_u$ *with the model parameter* $\phi$*.*

$$q_\phi(z_u|x_u).$$

[2] $x_{u,i} \in \{0, 1\}, x_{u,i} = 1$ representing an interaction between user $u$ and item $i$, n is the item num.

As Figure 2 (a) shows, The fundamental assumption of the problem is that the users themselves (**U**) solely influence the user preferences in the feedback data (**P**). Numerous outstanding approaches have arisen to address the above problem, and their solutions can be uniformly summarised into one class of solutions, i.e., for a user, they assume the observed data is generated from the following distributions:

$$p_\theta(x_u) = E\left[\int p_\theta(x_u|z_u)p_\theta(z_u)dz_u\right]. \tag{1}$$

Such approaches will restore the generation process of the data with outstanding performance, but they overlook the influence of confounders on feedback in user interactions. As a result, they cannot explain why the user feedback on ice cream shows opposite results due to wage, where wage acts as a confounder. Based on the above findings, we naturally consider incorporating confounders into the data generation process as a more reasonable modeling approach. Firstly, we redefine the problem as follows:

Problem 2. *Given the historical interaction data $x_u = \{x_{u,1}, ..., x_{u,n}\}$, we can predict the user preference $z_u, c_u$ with the model parameter $\phi$.*

$$q_\phi(z_u, c_u|x_u).$$

A graphical representation is depicted in Figure 2 (b), illustrating that the inner preferences of users and other confounders influence the observed user preferences in feedback data. We must emphasize that we are exclusively considering unobserved confounders in this problem. For observable confounders, we prefer incorporating them as additional input to enhance the performance of models, such as POI information in POI recommendations. In a similar vein, we propose a feasible solution to the aforementioned problem as follows:

$$p_\theta(x_u) = E_{p_\theta(c_u)}\left[\int p_\theta(x_u|z_u, c_u)p_\theta(z_u)dz_u\right]. \tag{2}$$

By employing this approach, we can effectively disentangle user preferences from confounders, thus overcoming the limitations of conventional methods. For example, we can use the do-operation to predict user interactions in their current environment, enabling us to answer a counterfactual question such as "Will users prefer ice cream if the weather is warm?". Even when the labels of the relevant confounders are unknown, we can obtain purer representations of users' preferences than composite entities.

### 3.2 Causal Modeling of Unobserved Confounders

When users interact with items, they are inevitably influenced by confounders such as weather, location, etc. Hence, many models that utilize additional information, such as location data as supplementary input, often demonstrate better predictive performance. However, observable confounders represent only a small fraction of this vast population. In most cases, confounders are unobservable. Therefore, disentangling the impact of unobservable confounders from feedback data remains challenging. To address this issue, we start by making a mild assumption on the independence of the confounders and give corresponding proofs:

Assumption 1. *Given confounders $\mathbf{C} = \{c_1, c_2, ..., c_k\}$, we assume that is independent of the user.*

$$c_j \perp u_i, \quad i \in [1, m], j \in [1, k].$$

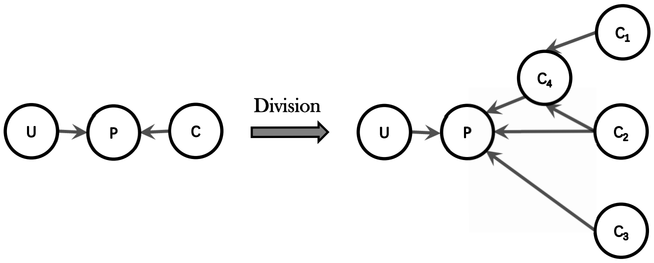


Fig. 3. An example showing confounders disentanglement in latent space, where the confounders **C** are further dismantled to k concepts $\{c_1, c_2, ..., c_k\}$ based on the given causal graph.

One premise for this assumption is that the data collection environment for user feedback is consistent. This condition is often easily met in recommender systems, where data is sourced from historical interactions within a specific platform over a certain period. Hence, this assumption is reasonably mild and applies to most recommendation scenarios. With this assumption, we can separate the confounders from the inherent preferences of users, thus achieving the model architecture as depicted in Eq 2.

Next, we will provide the corresponding proof of Assumption 1:

Proof. In our paper, the confounder set **C** includes various unobserved confounders, such as wage. The representation of confounders is not dependent on specific users; thus, the exogenous variables of **C** are not dependent on the exogenous variables of **U**. Similarly, the user's inherent preference **U** will not change due to the point of interest (poi) or other confounders. Thus, the exogenous variables of **U** are not dependent on the exogenous variables of **C**. Given the exogenous variables $E_U \rightarrow \mathbf{U}$ and $E_C \rightarrow \mathbf{C}$:

$$\mathrm{E_U} \perp \mathrm{E_C}.$$

In the causal graph shown in Figure 2 (b), **U→P←C** is a collider. According to the characteristics of colliders discussed in Section 2.3 of Pearl's book [16], when the condition of independence of the exogenous variables of **C** and **U** is met, **C** and **U** are independent. □

One of the constraints for the validity of Assumption 1 is that the exogenous variables of the confounders must be independent of the users, which means our model can only account for confounders that do not rely on the user, such as location, weather, etc. However, confounders closely related to the user, such as social relationships and background, cannot be included in our model.

. From this, it can be inferred that the assumption of independence between **C** and **U** in the paper is reasonable. In our model, the representations of confounders and user preferences are extracted using different neural networks through feedback data, satisfying the independence assumption mentioned above. Additionally, based on the characteristics of the collider, we can infer that when we condition the preference in the training set, **C** and **U** are likely to be dependent. This observation explains the intuition that **C** and **U** are correlated and provides theoretical evidence for this correlation. We have noted this characteristic and made special design considerations in our methodology to address this aspect. Thus, we did not directly use the representation of confounders for modeling (randomly initialized embedding representation of confounders in [35, 36, 38]), but instead employed local and global causal graphs to capture this correlation (details in Section 3.3).

Moreover, the relationships among confounders are not independent and follow a causal structure represented by a specific causal graph. For instance, consider two confounders: location and item category. The category of items depends on the nature of the location (e.g., *location* → *item category*); for example, clothes will be sold in a shopping mall and will not appear in a library. To formalize the causal representation, we consider $k$ confounders in the data. The confounders are causally structured by a Directed Acyclic Graph (DAG) with an adjacency matrix **A**. For convenience, we adopt a linear Structural Causal Model (SCM) as previous research [35, 36] to model the relationship between confounders and the causal graph, as illustrated in Eq 3:

$$\mathrm{C} = \mathrm{A}^{\top}\mathrm{C} + \epsilon = (I - \mathrm{A}^{\top})^{-1}\epsilon, \tag{3}$$

where **A** is the parameter to be learned by our model and is initialized by the all-one matrix, $\epsilon$ is independent Gaussian noise acting as the exogenous variables of the confounders, and **C** is a structured causal representation of the $k$ confounders generated by a DAG. This approach can further disentangle the confounders based on the causal graph **A**, as depicted in Figure 3. As expected, nonlinear SCM is more suitable for complex scenarios like recommender systems than linear ones. Therefore, in our practical deployment, we utilize the nonlinear SCM, which will be further elucidated in Section 3.3.

### 3.3 Causal Structure Learning

As mentioned before, an accurate causal graph enables us to better capture the influence of confounders on user preferences and the dependencies among these confounders. In conventional causal graphs, a causal flow between nodes is typically represented by an adjacency matrix, and the weights in this matrix measure the influence of parent nodes on their respective child nodes. However, when applied in recommendation system scenarios, this approach falls short of capturing the heterogeneity among users.

Consider two different users in the context of music recommendation. One user enjoys listening to different types of music in different locations (e.g., quiet music in the library), while the other prefers the same type of music regardless of location. The impact of location on these users differs: for the former, location influences their music preferences, while the latter remains unaffected by it. Although the location-music causal relationship can be inferred from the global causal graph, using this global graph alone to predict the impact of location on the music preferences of these two users would lead to suboptimal recommendations for the latter. To address this issue, we employ a combination of global and local causal graphs to capture the influence of confounders on user preferences. The global causal graph captures as many causal relationships as possible within the given constraints, and the local causal graph optimizes recommendation performance by masking the irrelevant relations of the global graph that do not impact the current user.

*3.3.1* ***Global SCM.*** Specifically, we use the global causal graph to model the relationships among all confounders. Given the adjacency matrix $\mathcal{G}^{global}$, it is associated with the true causal graph. In this context, $\mathcal{G}^{global}_{ij}$ can be viewed as an indicator vector, where $\mathcal{G}^{global}_{ij} = 1$ signifies that node $i$ is the parent node of node $j$, indicating that node $j$ is influenced by node $i$. In contrast, $\mathcal{G}^{global}_{ij} = 0$ implies that node $j$ and node $i$ are unrelated. The global causal graph is primarily used to capture dependencies among confounders without focusing on the strength of the dependencies between any two dependent confounders.

Therefore, we only require a binary adjacency matrix to meet this need. We begin by initializing an all-one learnable adjacency matrix $\mathcal{G}^{global}$, which is subsequently binarized to meet the specified requirements. To make the binary

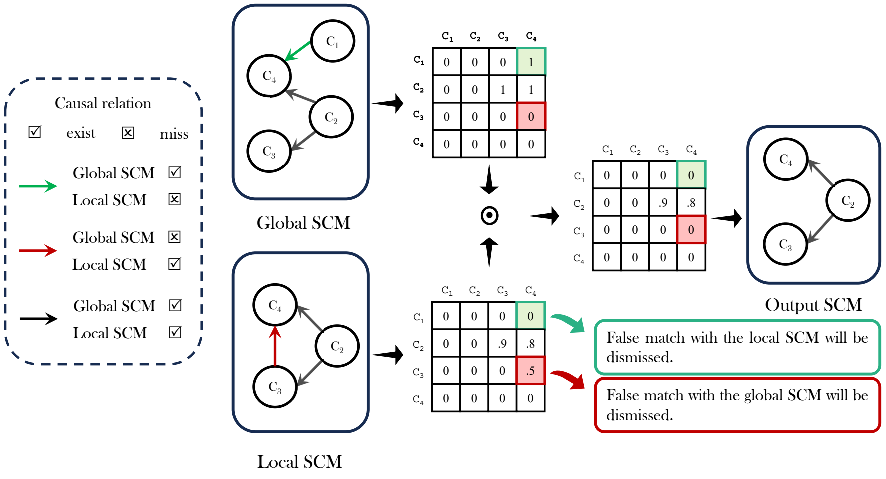


Fig. 4. Example of a Global and Local Causal Structure.

operation continuous, we leverage the $Gumbel\text{-}Softmax$ to get the binary adjacency matrix, which gives a continuous approximation to sampling from the categorical distribution [13–15]. We adopt a similar approach by adding Gumbel noise to the sigmoid function, which we formula as *Gumbel-Sigmoid*:

$$Gumbel\text{-}Sigmoid(\mathcal{G}^{global}) = \frac{exp((\mathcal{G}^{global} + \dot{g})/\tau)}{exp((\mathcal{G}^{global} + \dot{g})/\tau) + exp(\ddot{g}/\tau)}, \tag{4}$$

where $\dot{g}$ and $\ddot{g}$ are two independent Gumbel noises, and $\tau \in (0, \infty)$ is a temperature parameter. As $\tau$ diminishes to zero, a sample from the *Gumbel-Sigmoid* distribution becomes cold and resembles the one-hot samples. Our experiments show that a small fixed $\tau$ (e.g., 0.2) works well.

*3.3.2* ***Local SCM.*** Once we obtain a usable global causal graph, we use local causal graphs to measure the strength of dependencies between confounders. The VAE-based models usually obtain the user preference representation in latent space from the user feedback data by encoders, and our model follows this process to obtain the mixed preference in feedback data:

$$z = \text{Encoder}(x_u), \tag{5}$$

where $z$ is the representation of mixed preference in feedback data. To separate the confounders from $z$, we use a nonlinear function to obtain the projection of $z$ in the space of confounders, which can be formulated as follows:

$$c_i = f_i(z), \quad i \in [0, k], \tag{6}$$

where $c_i$ is the $i\text{-}th$ confounder and $f_i(\cdot)$ is the corresponding function. In practice, we use two linear layers to simulate. Then we can calculate the exogenous variables $\epsilon$ by transforming Eq. 3:

$$\epsilon = \{I - (\mathcal{G}^{global})^\top\}\text{C}. \tag{7}$$

In our model, the mapping function of confounders is shared among users. Therefore, to capture user specificity, we must introduce additional user-specific information to distinguish between users. The conventional approach involves using additional user embeddings as personalized user information. However, this leads to increased computational complexity and parameters in the model. To mitigate this drawback, we utilize a MLP layer to map the encoder output to obtain personalized user preference without significantly increasing the temporal and spatial complexity of the model,

$$sInfo_u = \mathsf{MLP}(z), \tag{8}$$

where $sInfo_u$ is the user-specific preference after MLP Layer, as Figure 5 shows. By incorporating additional user-specific information, we can obtain distinct representations of confounders for each user. Furthermore, we utilize an attention mechanism to calculate the strength of dependencies between confounders. In our experiments, we observed that the multi-head attention performs better as it effectively captures the heterogeneity among different confounders.

$$\mathcal{G}^{local} = \mathsf{MultiHead}(\epsilon, sInfo_u), \tag{9}$$

where the number of attention heads corresponds to the number of confounders. The *Multi-Head* layer is illustrated in Figure 5.

### 3.3.3 *Causal Layer.*

Given the global causal graph and local causal graphs, we perform calculations in the causal layer to obtain the final user-specific causal graph.

$$\mathcal{G}^{u} = \mathcal{G}^{global} \odot \mathcal{G}^{local}, \tag{10}$$

where $\odot$ is the element-wise multiplication. As Figure 4 shows, $\mathcal{G}^{u}_{ij} = 1$ if and only if both $\mathcal{G}^{\mathbf{global}}_{\mathbf{ij}} = \mathbf{1}$ and $\mathcal{G}^{local}_{ij} \neq 0$ hold, which reveals two reasonable potential conditions. Firstly, the local causal graphs must adhere to the global causal graph. This means that any two confounders without a causal relationship in the global causal graph cannot establish causality through the local causal graphs. The global causal graph represents the true causal graph; thus, any causal relationship absent in the global causal graph, even if present in the local causal graphs, is considered erroneous and disregarded. Secondly, any two confounders without a causal relationship in the local causal graphs cannot influence the current user through the global causal graph. The local causal graphs capture the influence of confounders on the user. If there is no causal relationship between two confounders in the local causal graph, this causal pathway cannot influence the user. Hence, including such pathways would affect the final performance and is therefore not considered. Once we obtain the final causal graph, we can derive reconstructed causal representations of the confounders according to Eq 3:

$$\begin{aligned} \hat{c}_i &= g_i(\mathcal{G}^{u}_{i} \odot \{I - (\mathcal{G}^{global})^{\top}\}^{-1}\epsilon), \\ \hat{\mathrm{C}} &= \{\hat{c}_1, \hat{c}_2, ..., \hat{c}_k\}, \end{aligned} \tag{11}$$

where $g_i(\cdot)$ is a mild nonlinear function ($sigmoid(\cdot)$ in our experiments), which is less sensitive to input variations, thereby facilitating more efficient parameter updates during the training process. For any confounder $c_i$, $\odot$ represents considering only the influence of its parent nodes, excluding the influence of other irrelevant nodes.

### 3.3.4 *Mask Layer.*

"do-operation" is a commonly used tool in causal theory through which you can tell us something about counterfactual problems, for example, ‘Will users prefer ice cream if the weather is warm?’. This mask layer can implement the "do-operation." We only need to provide an additional mask $\mathcal{G}^{mask}$, where $\mathcal{G}^{mask}_{ij} = 0$ indicates excluding the influence of node $i$ on node $j$ in the do-operation. Given the mask graph $\mathcal{G}^{mask}$, the reconstructed causal representations can derived from:

$$\begin{aligned} \mathcal{G}^{u\text{-}masked} &= \mathcal{G}^{u} \odot \mathcal{G}^{mask}, \\ \hat{c}_i &= g_i(\mathcal{G}^{u\text{-}masked}_{i} \odot \{I - (\mathcal{G}^{global})^{\top}\}^{-1}\epsilon). \end{aligned} \tag{12}$$

If no explicit $\mathcal{G}^{mask}$ as input, $\mathcal{G}^{mask}$ is set as an all-one matrix to enhance the causal relation flow between confounders.

*3.3.5* ***Identification of the Learned Graph.*** As shown in Figure 5, we will utilize a parametric model like Variational Autoencoder (VAE), combined with a $k \times k$ binary adjacency matrix, to fit the observed data. Unsupervised learning of the model might be infeasible due to the identifiability issue as discussed in [11, 18, 35]. To demonstrate the identifiability of the learned graph, we prove that under appropriate conditions, the computation described above can lead to the recognition of the hypergraph of the true graph. Consider a marginal distribution $P(\mathbf{C})$ induced by a Structural Equation Model (SEM) defined in Eq 3 with Directed Acyclic Graph (DAG) $\mathcal{G}$, and our SEM Eq 12 induces the same marginal distribution, where the binary adjacency matrix represents a DAG $\mathcal{H}$, we can obtain Lemma 1 if the $g_i(\cdot)$ is not a constant function and its proof.

Lamma 1. *$\mathcal{H}$ is a super-graph of $\mathcal{G}$, i.e., all the edges in $\mathcal{G}$ also exist in $\mathcal{H}$.*

Proof. First, let's consider the case where the $g_i(\cdot)$ is a constant, w.r.t. $c_j$ whether the $\mathbf{A}_{ji} = 0$ or $1$ do not affect $c_i$, but will change the causal graph $\mathcal{H}$, so when $g_i(\cdot)$ is constant, we can not uniquely identified the $\mathcal{H}$ from $P(\mathbf{C})$. Fortunately, in recommender systems, $g_i(\cdot)$ usually satisfies the non-constant condition. Then, we restrict $g_i$ to be non-constant, w.r.t. all $c_j$, $j \neq i$ to meet the causal minimality condition.

It suffices to show that if $c_j$ is not a parent of $c_i$ in $\mathcal{H}$, then $c_j$ is not a parent of $c_i$ in $\mathcal{G}$, either. That $c_j$ is not a parent of $c_i$ in $\mathcal{H}$ indicates $\mathbf{A}_{ji} = 0$. Therefore, $g_i(\mathbf{A}_i \odot \mathbf{C})$ is a constant function w.r.t. $c_j$. For the reduced SEM with functions $g_i$'s and causal DAG $\mathcal{G}$, we conclude that $c_j \notin c_{pa_i}$ and the input arguments of $g_i$ do not contain. Thus, $c_j$ cannot be a parent of $c_i$ in $\mathcal{G}$. □

As Peters' book [17][Theorem 27] shows, if the $P(\mathbf{C})$ is generated by a restricted additive noise model (ANM), the true causal graph is identifiable. Thus, we further assume a restricted ANM for the data-generating procedure to ensure the true causal graph $\mathcal{G}$ is identifiable. We then obtain the following proposition with proof.

**Proposition** 1. *Assume a restricted ANM with graph $\mathcal{G}$ and distribution $P(\boldsymbol{C})$ so that the original SEM is identifiable. If the parameterized SEM in the form of Eq. 12 with graph $\mathcal{H}$ induces the same $P(\boldsymbol{C})$, then $\mathcal{H}$ is a super-graph of $\mathcal{G}$.*

Proof. Recall that the reduced SEM with $g_i$'s and graph $\mathcal{G}$ satisfies the causal minimality condition and has the same distribution $P(\mathbf{C})$. With the identifiability result of restricted ANMs [17], we know that $\mathcal{G}$ is identical. Applying Lemma 1 completes the proof. □

Then, we can apply a parametric model and a binary adjacency matrix to fit the SEM in Eq. 3. Suppose the causal relationships fall into the chosen model functions, and we can obtain the exact solution that minimizes the negative log-likelihood given infinite samples. In that case, the resulting SEM has the same distribution [9]. Consequently, we obtain an acyclic supergraph from which existing nonlinear variable selection methods can be used to learn the parental sets and the causal graph.

*3.3.6* ***Mix Layer.*** With the reconstructed causal representation of confounders $\hat{\mathbf{C}}$ and user-specific preference, we can drive the mixed preference representation of feedback data in latent space. For $k$ confounders, a user may be influenced by some rather than all. To retain this characteristic, we take advantage of the attention mechanism to model the

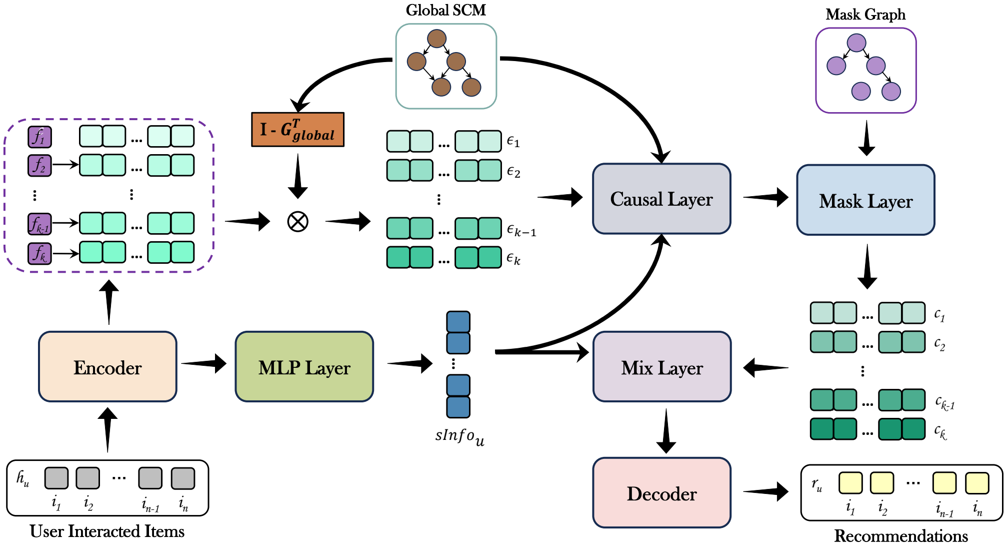


Fig. 5. The architecture of CSA-VAE.

confounder's influence on the user. Specifically, we calculate the attention score as follows:

$$Q = f(\mathrm{Norm}(sInfo_u)), \quad K = g(\mathrm{Norm}(\hat{\mathbf{C}})), \quad V = h(\hat{\mathbf{C}}),$$
$$score = \mathrm{Softmax}(\frac{QK^\top}{\sqrt{d}}), \tag{13}$$
$$\mathbf{C^u} = score \cdot V,$$

where $f$, $g$ and $h$ are learnable linear layers, $\mathrm{Norm}(\cdot)$ means normalization, $d$ is the latent embedding size, and $\mathbf{C^u}$ is the representation of confounders influence on user $u$. Then we mix the $sInfo_u$ and $\mathbf{C^u}$ to get the reconstructed mixed user preference representation of feedback data in latent space as follows:

$$\hat{z} = \mathrm{FFN}(sInfo_u + \mathbf{C^u}), \tag{14}$$

where $\mathrm{FFN}(\cdot)$ denotes the feed-forward layer, but in practice, we found simply add $sInfo_u$ and $\mathbf{C^u}$ is enough. The Mix layer combines the output of user-specific preferences with the output of the causal layer - the causal representations of confounders, resulting in a user representation under the influence of confounders. This user representation is then input into the decoder to reconstruct observed data.

## 4 INFERENCE AND LEARNING STRATEGY

The overall architecture of our model is depicted in Figure 5. This section describes the inference stage to make recommendations and the training stage to train our model to learn user preferences and causal graphs simultaneously.

### 4.1 Inference process

Our model CSA-VAE can explicitly model the unobserved confounders and user-specific preference, thus choosing whether to incorporate the influence of confounders.

*4.1.1* ***Recommendation with confounders.*** Given the reconstructed mixed user preference representation $\hat{z}$, we can get the predicted user-item relevance score as follows:

$$\{\hat{x_1}, \hat{x_2}, ..., \hat{x_n}\} = \mathsf{Decoder}(\hat{z}). \tag{15}$$

*4.1.2* ***Recommendation without confounders.*** We use *MaskLayer* to completely Shielding from the influence of confounders with the all-zero $\mathcal{G}^{mask}$, the simplified formula is as follows:

$$\{\hat{x_1}, \hat{x_2}, ..., \hat{x_n}\} = \mathsf{Decoder}(\mathsf{FFN}(sInfo_u)). \tag{16}$$

*4.1.3* ***User Controllable Recommendation.*** The causal graph and the learned representations of the confounders allow users to interactively adjust the representation of preferences of recommender systems learned from their past behavior to better align with their current preferences. In this paper, we formalize the task of user-controllable recommendation.

**Task Definition of User Controllable Recommendation.** Let $x_u = \{x_{u,1}, x_{u,2}, \ldots, x_{u,n}\}$ represent the historical interaction data of user $u$, where each $x_{u,i}$ corresponds to a specific past interaction between the user and an item. Let $z$ be the learned mixed preference representation of the user derived from their past behavior, and $c_u$ represent the latent confounders that may influence the user's preferences. The user-controllable recommendation task is designed in the recommendation model to allow users to adjust the learned preference representation $z$ to align their current intentions or preferences.

**Solution based on CSA-VAE.** Our model CSA-VAE incorporates confounders $c_u$ that may capture external influences on users' preferences. We model confounders as latent variables, and the user can adjust their influence interactively. Users can modify the confounder representations $c_u$ to better align with their current preferences as follows:

$$p_\theta(x_u) = E_{p_\theta(c_u)}\left[\int p_\theta(x_u|z_u, do(c_u^i), c_u^j) p_\theta(z_u) dz_u\right], i, j \in [1, k] \ \ and \ \ i \neq j, \tag{17}$$

where $do(\cdot)$ represents a causal intervention, meaning that the confounders $c_u^i$ are manipulated or set to specific values during the recommendation process. "do-operation" is a key feature of causal inference, allows us to model the influence of changing these confounders on the user's preferences. The ability to manipulate confounders in this way gives the user control over how these confounders influence their current mixed preferences and the generated recommendations.

### 4.2 Training process

*4.2.1* ***Evidence Lower Bound.*** Once we obtain the latent representation $\hat{z}$ of users' mixed preferences in the feedback data, we can reconstruct the observed data as follows:

$$p_\theta(x_u|\hat{z}) = p_\theta(x_u, \hat{z}|sInfo_u, \mathbf{C}). \tag{18}$$

We follow the variational autoencoder (VAE) paradigm [6] and optimize $\theta$ by maximizing the lower bound $\sum_u \ln p_\theta(x_u)$, where $\ln p_\theta(x_u)$ is bounded as follows:

$$\ln p_\theta(x_u) \geq E_{q(z,|x_u, sInfo_u, \mathbf{C})} \left[\ln p(x_u|z, sInfo_u, \mathbf{C})\right] - D_{KL}(q(z|x_u, sInfo_u, \mathbf{C})||p(z|sInfo_u, \mathbf{C})). \tag{19}$$

The relevant proofs are provided in Appendix A.1.

*4.2.2* ***Constraints of the Causal Graph.*** Causal inference requires discovering the underlying causal structure between variables. A DAG is commonly used to represent such causal relationships because it naturally encodes the assumption that causality flows in one direction and does not form loops. Thus, the causal adjacency matrix $\mathbf{A}$ is constrained to be a DAG. We employ a continuous distinguishable constraint function instead of the traditional combinatorial DAG constraint [36]. This function attains zero if, and only if the adjacency matrix $\mathbf{A}$ corresponds to a DAG [36]:

$$\mathbf{H}(\mathbf{A}) = tr((I + \frac{c}{k}\mathbf{A} \odot \mathbf{A})^k) - k = 0, \tag{20}$$

where c is an arbitrary positive number, controls the spectral radius of the graph, and ensures that the adjacency matrix does not lead to significant or unstable eigenvalues. $k$ is the number of confounders. The value of $c$ is the spectral radius of $\mathbf{A}$, and due to nonnegativity, it is bounded by the maximum row sum by the Perron-Frobenius theorem.

#### 4.2.3 *Diversity constraint of confounders.*

To ensure the diversity and comprehensiveness of confounders, we incorporate a similarity penalty in the loss function to guide the diversity of parameters. This method effectively prevents the model from focusing too narrowly on specific patterns, thereby enhancing its generalization capability. By employing this penalty mechanism, the model can explore a broader parameter space during training, resulting in more comprehensive and diverse representations of confounders. The formula is as follows:

$$\mathcal{L}_{sim} = \sum_{i}^{k}\sum_{j}^{k} \textit{Cosine-Similarity}(\mathrm{Norm}(\epsilon_i), \mathrm{Norm}(\epsilon_j)) \quad (i \neq j). \tag{21}$$

#### 4.2.4 *Objective function.*

The training procedure of our model reduces to the following constrained optimization:

$$\begin{aligned} maximize \quad & \ln p_\theta(x_u) \geq E_{p_\theta(\mathbf{C})}\left\{E_{q_\theta(z_u|x_u,\mathbf{C})}[\ln p_\theta(x_u|z_u,\mathbf{C})] \right. \\ & \left. - D_{KL}(q_\theta(z_u|x_u,\mathbf{C}) \| p_\theta(z_u)) \right\}, \\ s.t. \quad & (20), \\ s.t. \quad & (21). \end{aligned} \tag{22}$$

By the lagrangian multiplier method, we have the new loss function:

$$\mathcal{L} = -\mathbf{ELBO} + \mathbf{H}(\mathcal{G}^{global}) + \mathbf{H}(\mathcal{G}^{local}) + \mathcal{L}_{sim}. \tag{23}$$

## 5 EXPERIMENTS

In this section, we present the extensive experiments conducted on one semi-simulated dataset and five real-world datasets to demonstrate the effectiveness of the proposed CSA-VAE, with an emphasis on answering the following research questions:

- **RQ 1**: Can CSA-VAE obtain a useful causal graph of confounders without relevant labels? How does the strength of the causal relationship obtained compare to the true value?
- **RQ 2**: Can CSA-VAE achieve better performance compared to other baselines? How about the performance of CSA-VAE using only user preferences for recommendations?
- **RQ 3**: How do the causal relationships of confounders enhance the model's performance?
- **RQ 4**: How do the number of confounders influence the performance of CSA-VAE? Is the greater the number of confounders, the better?

| Dataset | Type | #Interactions | #User | #Items | Sparsity |
|---|---|---|---|---|---|
| Coat | Full-observed | 11,600 | 290 | 300 | 86.67% |
| Reasoner | | 58,497 | 2,997 | 4,672 | 99.58% |
| Yahoo!R3 | | 365,704 | 15,400 | 1,000 | 97.63% |
| Kuairec | | 12,530,806 | 7,176 | 10,728 | 83.72% |
| Epinions | Normal | 188,478 | 116,260 | 41,269 | 99.99% |
| ML-100k | | 100,000 | 943 | 1,682 | 93.69% |
| ML-1M | | 1,000,209 | 6,040 | 3,706 | 95.54% |
| ML-10M | | 10,000,054 | 69,878 | 10,677 | 98.66% |
| ML-20M | | 20,000,263 | 138,493 | 26,744 | 99.46% |
| ML-25M | | 25,000,096 | 162,542 | 59,048 | 99.74% |

Table 1. Statistics of the datasets

### 5.1 Dataset

It is difficult to verify the effectiveness of the causal graphs learned by CSA-VAE as the information of confounders is unobserved in the real dataset. Thus, we first validate the effectiveness of causal graphs on synthetic datasets and subsequently evaluate the recommendation performance of CSA-VAE on real-world datasets.

*5.1.1* ***Synthetic Dataset.*** We conduct experiments on synthetic data generated by the following process: We first assume that users are influenced by four confounders, where the exogenous variables for each confounder are generated by sampling from Gaussian distributions with mean and variance sampled from uniform distributions $[-3, 3]$ and $[0.01, 4]$, respectively. The intrinsic preferences of users are sampled from a standard normal distribution. Given the user's preference value $\mathcal{U}$, we obtain the user's personalized weights $w$ by sampling from a Poisson distribution. We generate samples using the causal structure model as shown in Eq 2. Finally, we input the confounders and user preferences into a two-layer MLP to generate the final observed value $\mathcal{X}$. Additional details and a formal description can be found in the Appendix A.2.

*5.1.2* ***Real-World datasets.*** To comprehensively and fairly validate the effectiveness of the model, we conducted experiments using nine publicly available datasets that encompass a variety of recommendation scenarios (such as movies and clothes) and different densities. Coat, YahooR3, Reasoner[3] [1], and Kuairec[4] [3] have fully observed data as the test set with the ground-truth relevance information. Following prior works, we binarize the ratings in YahooR3 and Coat by setting ratings ≥ 4 to 1 and the rest to 0. For Reasoner, we set ratings ≥ 4 to 1 and the rest to 0 as the regular train-test set and use the true user preference label like-unlike as the test set for evaluating the performance of models in capturing real user preference. For Kuairec, we use the sparse dataset for the train and the dense dataset for the test; the rating is binarized based on the ratio of user watching ratio. Specifically, setting watching ratio ≥ 2 to 1 and the rest to 0. We select five datasets of varying sizes ranging from 100k to 25M: ML-100K, ML-1M, ML-10M, ML-20M and ML-25M collected from the MovieLens website[5] to validate the robustness of the model to the dataset size. Additionally, we leverage the Epinions dataset, which originates from Epinions.com, a website where users can write reviews on

[3] https://reasoner2023.github.io/
[4] https://kuairec.com/
[5] https://grouplens.org/datasets/movielens/

various products and services and also rate the reviews written by other users. Following prior works, [5, 22], we remove the "inactive" users who interact with fewer than 20 items and the "unpopular" items who have interacted with users less than 10 times. We split the dataset into 70% for training, 20% for testing, and the remaining for validation. All user ratings greater than or equal to four are set to 1, while the rest are set to 0.

*5.1.3* ***Baselines***. We compare our method with the corresponding base models and the state-of-the-art de-confounding methods that can alleviate the confounding bias in recommender systems in the presence of unobserved confounders.

- **MF** [8]: MF is a popular technique used in recommendation systems to predict user preferences for items. It is particularly effective for collaborative filtering
- **Multi-VAE** [10]: Variational autoencoders (VAEs) to collaborative filtering for implicit feedback with VAE.
- **Muti-DAE** [10]: variational autoencoders (VAEs) to collaborative filtering for implicit feedback with DAE.
- **Macrid-VAE** [13]: Achieves macro disentanglement by inferring the high-level concepts associated with user intentions while simultaneously capturing a user's preference regarding the different concepts.
- **Rec-VAE** [19]: RecVAE introduces several novel ideas to improve Mult-VAE.
- **CDAE** [27]: A novel method for top-N recommendation that utilizes the idea of Denoising Auto-Encoders.
- **InvPref** [26]: InvPref assumes the existence of multiple environments as proxies of unmeasured confounders and applies invariant learning to learn the user's invariant preference.
- **IDCF** [37]: A general de-confounded recommendation framework that applies proximal causal inference to infer the unmeasured confounders and identify the counterfactual feedback with theoretical guarantees.

CSA-VAE focuses on learning a causal graph without the use of relevant labels. To ensure a fair comparison, we do not include models that incorporate additional information (e.g., movie names, categories), such as models SEM-MacridVAE [25], CaD-VAE [23] and PlanRec [38].

## 5.2 Experimental Settings

*5.2.1* ***Setups.*** We implement CSA-VAE and baselines in PyTorch. All models are trained with the Adam optimizer via early stopping at patience = 10. We set the learning rate to 1e-3 and the $l_2$-regularization weight to 1e-6. For CSA-VAE, we tune the hyper-parameter concepts $k$ in the range of $[1, 2, 4, 8, 16, 32]$ for different datasets. To detect significant differences in CSA-VAE and the best baseline on each dataset, we repeated their experiments five times by varying the random seeds. We choose the average performance to report. All ranking metrics are computed at a cutoff K = $[10, 30]$ for the Top-$k$ recommendation. Our implementation of the baselines is based on the original paper or the open codebase Recbole [40].

*5.2.2* ***Evaluation Metrics.*** Note that the sampling-based evaluation approach does not truly reflect the ability of the model to capture the true preferences of users. Simply fitting the data may also have better performance. To this end, we report the all-ranking performance w.r.t. two widely used metrics: Recall and NDCG cut at K = $[10, 30]$. To measure the popularity of recommended items, we use average popularity rank (AVP) as the other indicator, and the formula is:

$$\text{Average Popularity (AVP)@K} = \frac{1}{|U|}\sum_{u \in U} \frac{\sum_{i \in R_u} \phi(i)}{|R_u|},$$

where $\phi(i)$ is the ascending order of times item $i$ has been rated in the training set, $\mathbf{R}_u$ is the recommended list of items for user $u$.

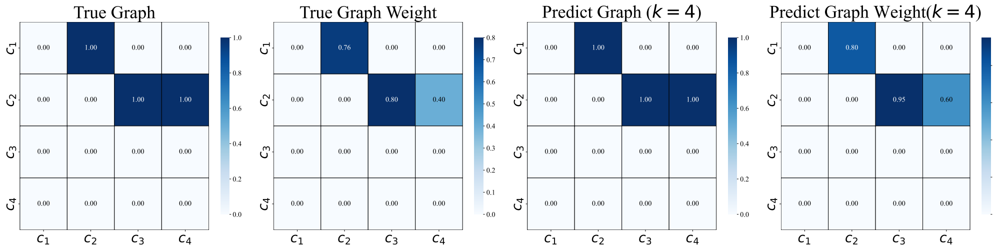


Fig. 6. In the graph prediction experiment on the synthetic data, the true graph (left) and the predicted graph (right).

| Datasets | Metric | K | MF | CDAE | Multi-DAE | Multi-VAE | Macrid-VAE | Rec-VAE | InvPref | ICDF | CSA-VAE | Imp.(%) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Coat | Recall ↑ | 10 | 0.03241 | 0.03418 | 0.03530 | 0.03461 | 0.03328 | 0.03539 | 0.03332 | 0.03439 | **0.03664** | 3.81% |
| | | 30 | 0.09966 | 0.10180 | 0.09815 | 0.09909 | 0.09810 | 0.10330 | 0.09806 | 0.09684 | **0.10647** | 3.07% |
| | NDCG ↑ | 10 | 0.05126 | 0.05362 | 0.05834 | 0.05534 | 0.05256 | 0.05665 | 0.05184 | 0.05022 | **0.05916** | 1.41% |
| | | 30 | 0.07871 | 0.08073 | 0.08212 | 0.08013 | 0.07846 | 0.08324 | 0.07776 | 0.07475 | **0.08557** | 2.79% |
| Yahoo!R3 | Recall ↑ | 10 | 0.03225 | 0.04493 | 0.06424 | 0.06560 | 0.02066 | 0.04648 | 0.02548 | 0.02519 | **0.06768** | 3.17% |
| | | 30 | 0.07687 | 0.10290 | 0.15280 | 0.15310 | 0.04693 | 0.10200 | 0.06069 | 0.05899 | **0.16169** | 5.61% |
| | NDCG ↑ | 10 | 0.01707 | 0.02440 | 0.03069 | 0.03149 | 0.01121 | 0.02503 | 0.01330 | 0.01372 | **0.03480** | 10.51% |
| | | 30 | 0.02893 | 0.03996 | 0.05387 | 0.05447 | 0.01846 | 0.03999 | 0.02275 | 0.02265 | **0.05928** | 8.83% |
| Reasoner | Recall ↑ | 10 | 0.00338 | 0.00234 | 0.00268 | 0.00356 | 0.00194 | 0.00386 | 0.00277 | 0.00276 | **0.00508** | 31.61% |
| | | 30 | 0.00959 | 0.00741 | 0.00961 | 0.01136 | 0.00723 | 0.00978 | 0.00926 | 0.00972 | **0.01445** | 27.20% |
| | NDCG ↑ | 10 | 0.00191 | 0.00129 | 0.00152 | 0.00165 | 0.00111 | 0.00193 | 0.00157 | 0.00155 | **0.00274** | 41.97% |
| | | 30 | 0.00366 | 0.00273 | 0.00343 | 0.00381 | 0.00261 | 0.00355 | 0.00339 | 0.00333 | **0.00488** | 28.08% |
| KuaiRec | Recall ↑ | 10 | 0.07115 | 0.06744 | 0.06826 | 0.06296 | 0.06382 | 0.06570 | 0.06939 | 0.06672 | **0.07442** | 4.59% |
| | | 30 | 0.09671 | 0.12428 | 0.12864 | 0.12455 | 0.12306 | 0.12640 | 0.12489 | 0.11964 | **0.13104** | 1.87% |
| | NDCG ↑ | 10 | 0.44834 | 0.43268 | 0.45814 | 0.43918 | 0.44796 | 0.44110 | 0.46040 | 0.43738 | **0.48594** | 5.55% |
| | | 30 | 0.27466 | 0.35317 | 0.36334 | 0.35451 | 0.35196 | 0.35164 | 0.34062 | 0.32359 | **0.36743** | 4.03% |

Table 2. The overall performance comparison results of applying our model and baselines on four real-world full-observed datasets. We evaluated the recommendation performance as a ranking task, underlined the best baseline result in each line, and put the best result in each line in bold; Higher Recall and NDCG mean better model performance. The arrow '↑' (or '↓') denotes that the higher (or lower) value means better performance on the metric. The 'Imp.' row reports the relative improvement or decline of CSA-VAE against the best baseline. The result is calculated based on the mean of five repetitions with different random seeds for all models on each metric.

### 5.3 Performance on the Synthetic Dataset (RQ1).

CSA-VAE can obtain a useful causal graph of confounders without relevant labels. The synthetic data set we used contained four confounders, resulting in a power of $2^{k(k-1)}$ possible relationships. Although the number of categories is not extensive, it still poses a challenging task. By the causal relationships between confounders present in the synthetic data, we can unambiguously determine the ability of CSA-VAE to capture the causal relationships between confounders. Due to the strong correlation between the local graph and users, we only present the global graph obtained by the CSA-VAE here. As shown in Figure 6, the global graph learned by CSA-VAE is well aligned with the ground truth graph, thus demonstrating the ability of CSA-VAE to effectively capture the causal relationships between confounders. It is important to emphasize that we used the *Gumbel-Sigmoid* shown in Eq. 4, resulting in an approximation of a binary causal graph by CSA-VAE. The final experimental results strongly support this approach.

| Datasets | Metric | K | **MF** | **CDAE** | **Multi-DAE** | **Multi-VAE** | **Macrid-VAE** | **Rec-VAE** | **InvPref** | **ICDF** | **CSA-VAE** | **Imp.(%)** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **ML-100K** | Recall ↑ | 10 | 0.02393 | 0.04412 | 0.05554 | 0.05620 | 0.02259 | 0.05396 | 0.01344 | 0.01292 | **0.06207** | 10.44% |
| | | 30 | 0.06294 | 0.09724 | 0.15466 | 0.15226 | 0.05641 | 0.12149 | 0.03476 | 0.03425 | **0.15948** | 3.12% |
| | NDCG ↑ | 10 | 0.03067 | 0.04571 | 0.05168 | 0.05200 | 0.02299 | 0.05588 | 0.04518 | 0.03517 | **0.05876** | 5.15% |
| | | 30 | 0.04358 | 0.06497 | 0.08830 | 0.08715 | 0.03515 | 0.07879 | 0.03246 | 0.04245 | **0.09307** | 5.40% |
| **ML-1M** | Recall ↑ | 10 | 0.02665 | 0.02825 | 0.02903 | 0.02843 | 0.2726 | 0.02739 | 0.02926 | 0.02997 | **0.03017** | 0.67% |
| | | 30 | 0.06525 | 0.07540 | 0.07647 | 0.08782 | 0.07729 | 0.07762 | 0.07316 | 0.06961 | **0.08906** | 1.41% |
| | NDCG ↑ | 10 | 0.03373 | 0.03732 | 0.03054 | 0.02954 | 0.02372 | 0.03733 | 0.03515 | 0.03425 | **0.03855** | 3.27% |
| | | 30 | 0.05079 | 0.05497 | 0.05268 | 0.05235 | 0.05476 | 0.05593 | 0.05503 | 0.05289 | **0.05589** | -0.07% |
| **ML-10M** | Recall ↑ | 10 | 0.03101 | 0.03629 | 0.03552 | 0.03595 | 0.03565 | 0.3539 | 0.03482 | 0.03412 | **0.04785** | 31.85% |
| | | 30 | 0.06686 | 0.08193 | 0.12127 | 0.11968 | 0.11200 | 0.11933 | 0.07939 | 0.07917 | **0.13134** | 8.30% |
| | NDCG ↑ | 10 | 0.03610 | 0.04001 | 0.03374 | 0.03410 | 0.03725 | 0.03250 | 0.03834 | 0.03866 | **0.04506** | 12.62% |
| | | 30 | 0.04763 | 0.05579 | 0.06500 | 0.06456 | 0.05780 | 0.06334 | 0.05346 | 0.05407 | **0.07544** | 16.06% |
| **ML-20M** | Recall ↑ | 10 | 0.03462 | 0.03726 | 0.03802 | 0.03974 | 0.03445 | 0.03660 | 0.03738 | 0.03714 | **0.04064** | 2.26% |
| | | 30 | 0.07543 | 0.08559 | 0.12024 | 0.11965 | 0.11412 | 0.11681 | 0.08554 | 0.08564 | **0.12260** | 1.96% |
| | NDCG ↑ | 10 | 0.03891 | 0.04300 | 0.03746 | 0.03964 | 0.03631 | 0.03496 | 0.04083 | 0.04215 | **0.04534** | 5.44% |
| | | 30 | 0.05232 | 0.05952 | 0.06803 | 0.06904 | 0.06567 | 0.06537 | 0.05718 | 0.05677 | **0.07341** | 6.33% |
| **ML-25M** | Recall ↑ | 10 | 0.02972 | 0.03194 | 0.03612 | 0.03471 | 0.03293 | 0.03356 | 0.03374 | 0.03399 | **0.04298** | 18.99% |
| | | 30 | 0.07140 | 0.07828 | 0.11434 | 0.11129 | 0.09572 | 0.11106 | 0.07969 | 0.08177 | **0.12220** | 6.87% |
| | NDCG ↑ | 10 | 0.03697 | 0.03869 | 0.03652 | 0.03458 | 0.03173 | 0.03332 | 0.03731 | 0.03551 | **0.04373** | 13.03% |
| | | 30 | 0.05136 | 0.05493 | 0.06637 | 0.06360 | 0.06553 | 0.06259 | 0.05296 | 0.05288 | **0.07331** | 10.46% |
| **Epinions** | Recall ↑ | 10 | 0.00897 | 0.01034 | 0.01765 | 0.01732 | 0.01191 | 0.01650 | 0.01346 | 0.01311 | **0.01932** | 9.46% |
| | | 30 | 0.01828 | 0.02505 | 0.03956 | 0.04067 | 0.01834 | 0.04195 | 0.03957 | 0.03835 | **0.04665** | 11.20% |
| | NDCG ↑ | 10 | 0.00630 | 0.00702 | 0.00822 | 0.00807 | 0.00648 | 0.00709 | 0.00613 | 0.00599 | **0.00887** | 7.91% |
| | | 30 | 0.00740 | 0.00842 | 0.01337 | 0.01355 | 0.01128 | 0.01332 | 0.00928 | 0.00967 | **0.01431** | 5.61% |

Table 3. The overall performance comparison results of applying our model and baselines on six real-world normal datasets. We evaluated the recommendation performance as a ranking task, underlined the best baseline result in each line, and put the best result in each line in bold; Higher Recall and NDCG mean better model performance. The arrow '↑' (or '↓') denotes that the higher (or lower) value means better performance on the metric. The 'Imp.' row reports the relative improvement or decline of CSA-VAE against the best baseline. The result is calculated based on the mean of five repetitions with different random seeds for all models on each metric.

Additionally, we found that the correlation strength between the confounders obtained by CSA-VAE is slightly higher than the true value. This occurs because the model treats the learning of causal relations as a binary classification task. Although the global causal graph is specialized for determining whether a causal relationship exists between two confounders, the local causal graph also plays a role in this function. To achieve binary classification, the model tends to exaggerate values to ensure accuracy, resulting in a slightly higher correlation strength between confounders with a causal relationship. A larger causal relationship helps the model better measure the impact of confounders on user preferences, thus aiding in more accurate modeling of confounders and user preferences.

### 5.4 Comparision with Baselines (RQ2)

The comparison between CSA-VAE and various baselines is shown in Table 2 and Table 3. The best results (compared across two classes) are shown in bold, and the runner-ups are underlined. In summary, we have the following observations:

**(1) Consistent superior recommendation performance.** The result demonstrates that the CSA-VAE model consistently outperforms the baselines regarding Recall and NDCG across various datasets and evaluation metrics. Specifically,

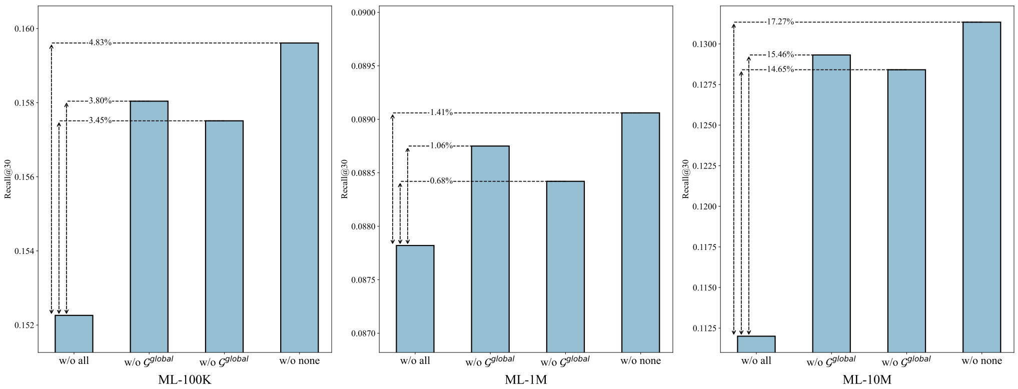


Fig. 7. Ablation experiments on global and local graphs conducted on Ml-100k, ML-1M and ML-10M with Recall@30.

CSA-VAE achieves the highest Recall and NDCG scores in nearly all cases, indicating its superior ability to recommend relevant items to users. Remarkably, CSA-VAE substantially improves Recall and NDCG compared to the baselines. The average performance boost against several state-of-the-art baselines achieves up to 9.55% across different datasets and evaluation settings, demonstrating the superiority of our model.

**(2) CSA-VAE can better model user preference.** As Table 2 shows, on unbiased datasets `Coat`, `YahooR3`, `Reasoner` and `Kuairec`, CSA-VAE user only user-specific preference for the recommendation, CSA-VAE achieves the highest Recall and NDCG scores in all cases, indicating its superior ability on model user preference rather than mixed preference in feedback data. Remarkably, CSA-VAE substantially improves Recall and NDCG compared to the deconfounded methods ICDF and Invpref, the average performance boost against several state-of-the-art baselines achieves up to 11.50%.

**(3) CSA-VAE can better fit the mixed preference in feedback data.** As Table 3 shows, on normal datasets `ML-100K`, `ML-1M`, `ML-10M`, `ML-20M` and `ML-25M`, and `Epinions`, CSA-VAE achieves the highest Recall and NDCG scores in all cases, indicating its superior ability on model user preference rather than mixed preference in feedback data. The slightly weaker performance on ML-1M compared to Rec-VAE is due to the fact that Rec-VAE uses additional training tricks, such as alternating updates of prior, which we did not use. The datasets with higher sparsity levels, such as ML-20M, Ml-25M, and Epinions, often present more challenges for traditional recommender systems due to the scarcity of user-item interactions. However, the CSA-VAE model demonstrates substantial improvements in these datasets, suggesting its robustness in handling sparse data and providing meaningful recommendations despite the challenges posed by data sparsity.

In summary, the comprehensive analysis of the performance of CSA-VAE across these datasets underscores its potential to significantly enhance recommendation quality and user engagement across a spectrum of real-world applications. The consistent outperformance over baseline models highlights the efficacy of integrating causal graph-based approaches in recommender systems, addressing unobserved confounders and providing more accurate and satisfying recommendations.

## 5.5 Ablation Study (RQ3).

### 5.5.1 *Effectiveness of Global and Local Graphs.*

The Figure 7 presents results for different variants involving both the global and local graphs:$w/o\ \mathcal{G}^{none}$ without both global and local graphs, equivalent to Multi-VAE; $w/o\ \mathcal{G}^{local}$

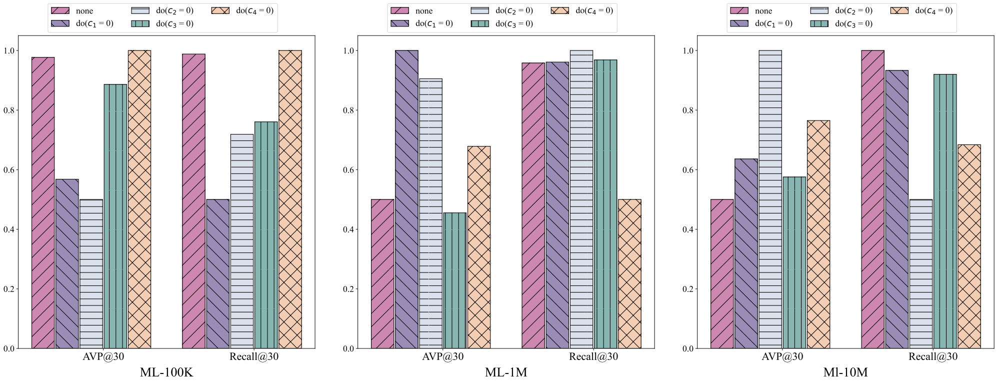


Fig. 8. "do-operation" experiments conducted on Ml-100k, ML-1M and ML-10M with Recall@30 and AVP@30, the number of confounders, $k = 4$. $do(c_i = 0)$ means performing the "do-operation" on $c_1$, $c_2$, $c_3$, and $c_4$, and $do(none)$ means no "do-operation" on any confounders.

(without local graph); $w/o\ \mathcal{G}^{global}$ (without global graph); and $w/o\ \mathcal{G}^{none}$ (with both global and local graphs). we have the following observations:

**(1)** Using either the global graph or the local graph alone results in improved performance on the dataset. The global graph captures the macro-level causal relationships among confounders, albeit losing specificity to users. On the other hand, the local graph captures user specificity but loses the accurate relationships between confounders. Both contribute partially to the causal relationships between confounders, leading to an enhancement in model performance.

**(2)** Simultaneously using the local and global graphs results in a performance improvement greater than the sum of their individual contributions. The global graph can correct the erroneous causal relationships between confounders in the local graph, while the local graph assigns user-specific weights to the global graph. The synergy of both significantly enhances the model's performance. In summary, the collaborative effect of both graphs significantly enhances recommendation quality by enabling a more comprehensive understanding of user behavior and preferences, creating a powerful model for accurate and effective recommendations.

*5.5.2* ***"do-operation" with Mask Graph.*** We conducted common "do-operation" experiments in the causal inference domain on the `ML-100K`, `ML-1M`, and `ML-10M` datasets, and the results are shown in Figure 8. We employed four concepts of confounders ($k = 4$) to model the unobserved confounders in these datasets. We used the popularity of items recommended (AVP@30) to the users and Recall@30 to evaluate the effect of the confounders. It is important to note that applying the "do-operation" to a confounder $c_i$ involves using an additional masking matrix and performing a dot product with the global graph. In the masking matrix, the $i$-th row is all zeros (indicating no influence as a parent node), and the rest of the rows are all ones. With this operation, we can answer the question: "If confounder $c_k$ has no effect, how does this impact user performance?" From the Figure 8, we have the following observation:

**(1) Confounder do harmful to user preference modeling.** When we operate on confounders, the performance of the model is mostly affected, which means that the confounders play an important role in the final recommendation process, where user-specific preferences should be absolutely dominant. The existence of confounders leads to the inevitable influence of confounders in the process of modeling user-specific preferences. Therefore, decoupling confounders and user-specific preferences is a necessary approach to better model user preferences.

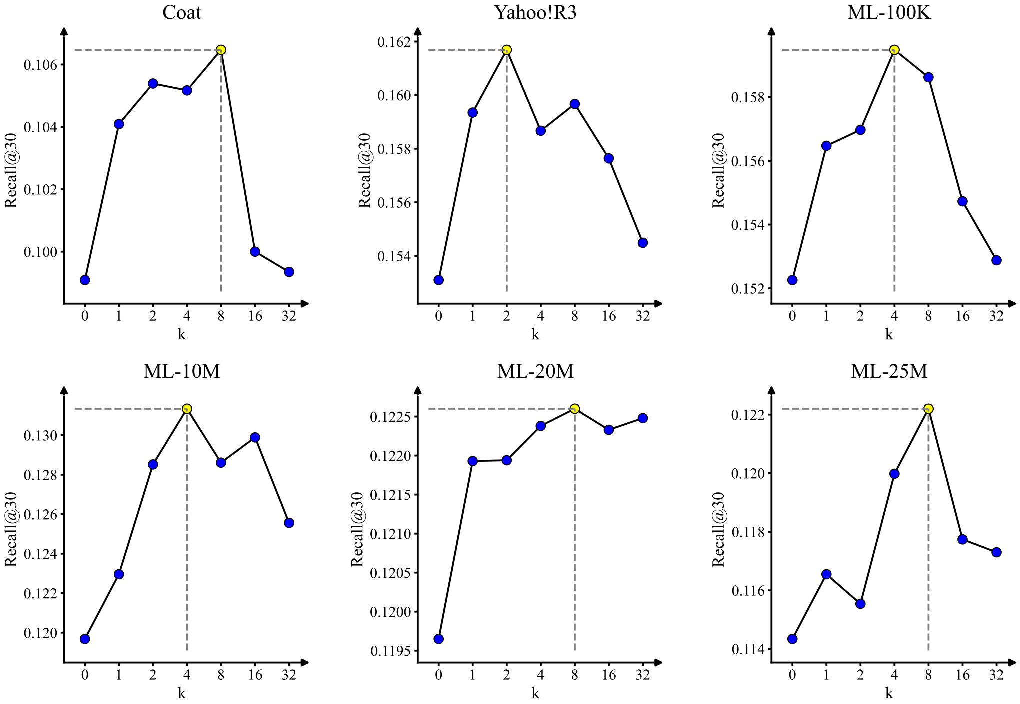


Fig. 9. Sensitivity of CSA-VAE with different confounders number $k$. The horizontal axes of all sub-figures are the variable $k$.

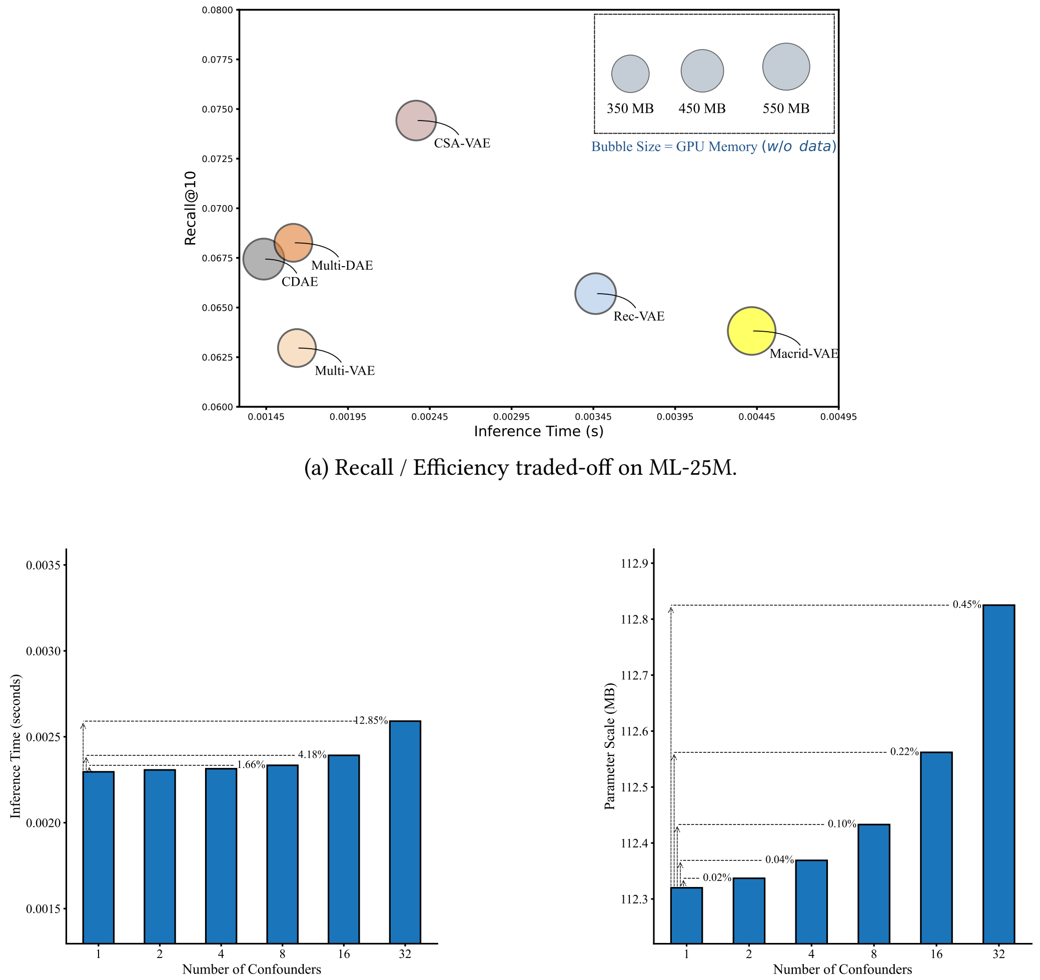
| **Model** | **CDAE** | **Multi-DAE** | **Multi-VAE** | **Macrid-VAE** | **Rec-VAE** | **CSA-VAE** |
|---|---|---|---|---|---|---|
| **Complexity** | $O(n(m+d))$ | $O(n(m+d))$ | $O(n(m+d))$ | $O(k{\cdot}n(m+d))$ | $O(n(m+d))$ | $O(n(m+k^2+d))$ |
| **FLOPs (M)** | 23.11 | 29.32 | 29.33 | 51.03 | 44.02 | 29.64 |

Table 4. Computational Complexity Comparison on ML-25M.

**(2) Confounder not totally harmful to user preference modeling.** When we operate on confounders on the ML-1M and ML-10M datasets, as Figure 8 shows, the popularity of recommend items increases, which means these confounders have a positive effect on mitigating popularity bias. Based on this finding, we can achieve the short-term goals of the recommendation system by controlling variables. For instance, we can recommend popular items to users at certain times while refraining from using additional recommendation strategies at other times.

### 5.6 Computational Complexity Analysis on ML-25M

We computed the model's Floating Point Operations (FLOPs) on the biggest dataset ML-25M using MMEengine [2] to indicate the model's learning complexity, which is a critical component in optimizing neural networks for performance and efficiency. The higher number of FLOPs signifies a more challenging learning process. The results are presented in Table 4 and Figure 10 (a). Additionally, we conducted experiments with varying numbers of confounders to validate



(a) Recall / Efficiency traded-off on ML-25M.

(b) CSA-VAE Inference Time

(c) CSA-VAE Parameter Scale

Fig. 10. Computational Complexity Comparison on ML-25M. (a) is the Recall / Efficiency traded-off comparison. A higher position on the vertical axis indicates better performance, while moving left along the horizontal axis signifies lower inference time costs. A smaller bubble size indicates reduced GPU memory costs during inference. (b) and (c) compare memory and inference time under various value numbers of confounders.

their effect on the model's inference cost on ML-25M. The corresponding results are shown in Figures 10 (b) and (c). Based on these experiments, we have the following observations:

- Compared to classical models, our model has a relatively lower increase in learning difficulty and outperforms other mainstream VAE-based models. Results shown in Table 4, introducing the causal graph as an additional learning task increases the learning difficulty of CSA-VAE. However, compared to Rec-VAE and Macrid-VAE, the increase in FLOPs is relatively lower.
- Our model balances model size and inference speed to a certain extent. As shown in Figure 10 (a), compared to mainstream VAE-based models (e.g., Rec-VAE), our model achieves higher inference speed and has significantly

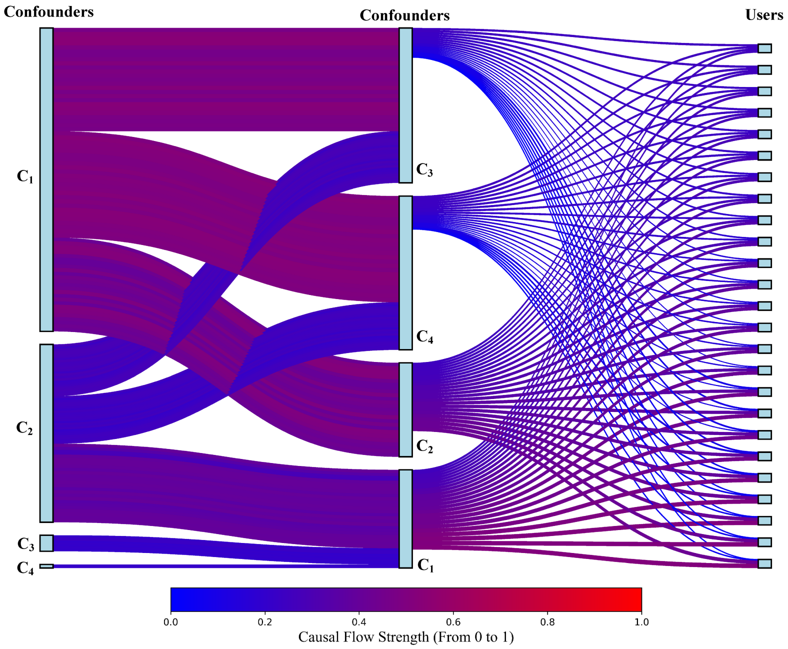


Fig. 11. Causal flow visualization based on Sankey (left to right). The left and middle columns represent the causal flow between confounders, and the right columns represent the causal flow between confounders and users (30 randomly selected). The color intensity indicates the strength of the causal relationship, and closer to red indicates more muscular strength (Since there is no intensity clipping, form weak loops (lower strength value) in the visualization).

smaller model parameters. Notably, compared to traditional models (e.g., CDAE), our model also shows a clear advantage in terms of parameter size.

- Under the condition of maintaining recommendation performance, the increase in model inference time caused by adding confounders remains within an acceptable range. As shown in Table 4, the number of confounders is a key determinant of the higher time complexity of CSA-VAE compared to other models, such as Multi-VAE. However, when the dataset is large, $m$ becomes much more extensive than $k$. At this point, the model's inference time is primarily influenced by the dataset size, with the number of confounders having a negligible impact. This observation is confirmed in Figure 10 (b). When $k = 8$, the inference time is only 1.66% higher than when $k = 1$, which falls within an acceptable range. It is also important to note, as shown in Figure 9, that tremendous values of $k$ do not yield better results. Therefore, a value of $k \leq 8$ is generally considered optimal.
- In our model, the number of confounders minimally impacts the model's parameter size. As shown in Figure 10 (c), when $k = 32$, the model size increases by only 0.45% compared to when $k = 1$, this increase in model size is almost imperceptible relative to the overall size of the model.

### 5.7 Causal Flow Visualization

We use Sankey to represent the flow and strength of causal relationships between confounders and between confounders and users. The results are shown in Figure 11, we have the following observations:

- The causal strengths between the confounders in the two columns on the left are not identical (the strength between the same two nodes varies from user to user), which should be the same if only using the global causal graph (without user information guidance), confirming that our model can generate user-specific causal graphs and capture complex nonlinear relationships between confounders under the guidance of user preference information.
- The causal strength exhibits clear user differences when it flows to users through confounders. The same confounder has varying levels of influence on different users; results demonstrate that our model captures the complex nonlinear relationships between confounders and user preferences.
- Users are affected to varying levels by different confounders, and the same user is affected to different capacities by multiple confounders, indicating that our model captures the varying sensitivity of users to different confounders.

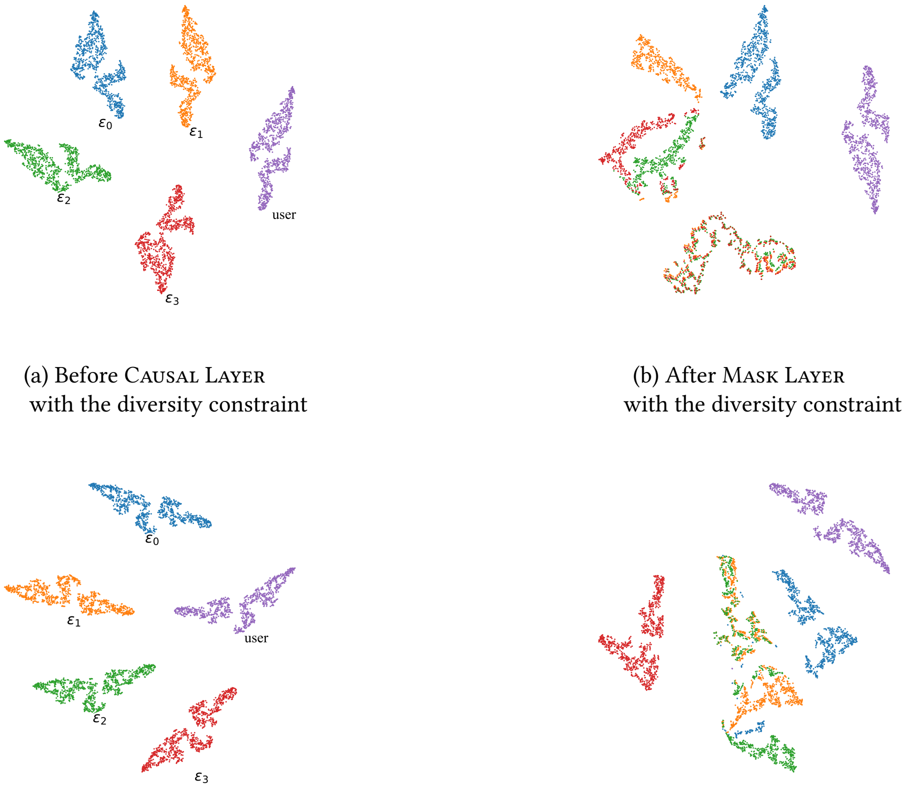
### 5.8 Sensitivity Analysis (RQ4).

We used various values of $k$ on the Coat, Yahoo!R3, ML-100k, ML-10M, ML-20M, and ML-25M datasets to verify the influence of the number of confounders. As Figure 9 (right side) shows, we have the following observations:
**(1) The performance of different numbers of confounders is closely related to the magnitude of the datasets.** As Figure 9 (right side) shows, With the increase in the number of confounders, the model performance decreases more on small data sets than on large data sets. This result arises because the dataset size is insufficient to fully support the model in identifying edges in the directed causal graph corresponding to the number of confounders. Specifically, for a directed causal graph with $k$ nodes, there are $2^{k(k-1)}$ possible edges, requiring at least $2^{k(k-1)}$ interactions to identify these edges reliably, and we need more due to the user-specific in the recommendation. Consequently, as the number of nodes increases significantly, inadequate data prevents the model from producing an improved graph, leading to a decline in performance.
**(2) The greater the number of confounders the better.** A larger $k$ implies a more diverse set of confounders. As $k$ increases, we obtain a finer-grained representation of the confounders, improving model performance. As Figure 9 shows, the model's performance increases with the increase of $k$ until reaching the critical point mentioned above. However, an excessively large $k$ may cause the learned representation of confounders to be more than the actual number of concepts influencing the data, leading to model overfitting and a subsequent decrease in performance.

### 5.9 Visualization of Confounders and User-specific preference

We use t-SNE [21] to visualize the confounders and user preference on ML-100K, with $k = 4$. Specifically, we first visual the exogenous variables of confounders and user preference before Causal Layer and then visual the representation of confounders and user preference after Mask Layer. We have the following observations:

- (1) As shown in Figure 12, the exogenous variables $\epsilon_k$ of $k$ confounders and user preferences exhibit five distinct clusters in both (a) and (c), confirming the independence between confounders and user preferences, thus supporting our Assumption 1.
- (2) After encoding global and local graphs, the causal relationship involving confounders manifests as adjacent clusters in the graph in both (b) and (d). Results demonstrate that our method CSC-VAE effectively captures causal relationships between confounders, thereby enhancing user-specific and mixed preferences modeling in feedback data.



(a) Before Causal Layer with the diversity constraint

(b) After Mask Layer with the diversity constraint

(c) Before Causal Layer without the diversity constraint

(d) After Mask Layer without the diversity constraint

Fig. 12. Visualization of confounders and user preference on ML-100K, $k = 4$. The same colors in (a)-(b) and (c)-(d) represent the same confounders. (a)-(c)is the visualization of exogenous variables $\epsilon_k$ and user preference before Causal Layer and (b)-(d) is the visualization of confounders and user preference after Mask Layer. The numbers $\{0, 1, 2, 3\}$ correspond to the elements in the confounders set $\{c_1, c_2, c_3, c_4\}$, and $\epsilon_i$ $(i \in [0, k])$ identifies the exogenous variables of the confounders. Among them, (a)–(b) are the results of training under diversity constraints, while (c)–(d) are the results of training without diversity constraints.

- (3) Regardless of the presence of diversity constraints, the exogenous variables of the confounders in (a) and (c) form independent clusters, demonstrating that the model can capture the independence of the exogenous variables of the confounders, which is not a result of the diversity constraints. Additionally, from (b) and (d), it can be observed that the confounders after the mask layer integrated better when diversity constraints are applied, indicating that the presence of diversity constraints helps the model obtain a better representation of the confounders and uncover the causal relationships between them.

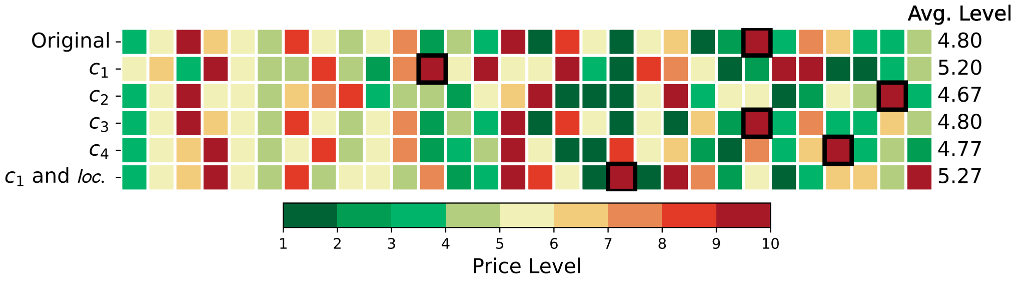


Fig. 13. The case study indicates the effectiveness of controlling confounders in recommender systems. We divided the prices in the `Epinions` dataset into ten equally sized intervals, with higher levels corresponding to higher prices. The selected user prefers high-priced items in his feedback data, and the black box represents the ground-truth item.

### 5.10 Case Study on Epinions

Using the `Epinions` dataset, which includes the price feature, we selected a user with a significant number of high-priced items in his historical interactions as the subject of the case study. By controlling for confounders, we generated alternative lists of recommended items for this user. We have the following observations:

- We can obtain a more accurate user recommendation list by controlling the confounders. As shown in Figure 13, controlling the confounder $c_1$ improves the average price level of the recommendation list and the ranking of the ground-truth item.
- Only the global causal graph is relied upon without using the local causal graph, achieving a suboptimal recommendation list. As shown in Figure 13, masking the local causal graph still improves the overall price level of the recommendation list, but the ranking of the ground-truth item remains suboptimal.
- When users are dissatisfied with the system's current recommendations, our model allows them to modify their recommendation list by controlling the confounders and causal graphs learned by the model. For example, when a user believes that the recommendation system's modeling expectations do not align with their preferences (e.g., a user sensitive to price), they can adjust the confounders to increase or decrease the average price level of the recommendations. Results demonstrate that our model, compared to traditional models, offers users greater flexibility, potentially improving overall user satisfaction.

## 6 CONCLUSION

Predicting user preferences in the presence of confounders is a challenging problem. We first redefined the problem, incorporating the influence of confounders into the model. We proposed a mild assumption to separate user preferences from confounders and used a combination of local and global graphs to capture the causal relationships between confounders and user-specific preferences. Finally, we proposed a VAE-based model called CSA-VAE. Extensive experiments are conducted on a synthetic dataset and nine real-world datasets to demonstrate the model's superiority. We theoretically proved the model's Evidence Lower Bound (ELBO) and the learned graph's identifiability. Furthermore, we employed the "do-operation" method to validate the controllability of the model, potentially offering users fine-grained control over the objectives of their recommendation lists with the learned causal graphs. Future work can explore advanced unsupervised clustering methods to obtain categories of confounders further, addressing the limitation of uncertainty in the impact categories of obtained confounders on user preferences.

## 7 ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of China No. 62472196, Jilin Science and Technology Research Project 20230101067JC, and the National Natural Science Foundation of China under Grant NO. 62176014, the Fundamental Research Funds for the Central Universities.

## REFERENCES

[1] Xu Chen, Jingsen Zhang, Lei Wang, Quanyu Dai, Zhenhua Dong, Ruiming Tang, Rui Zhang, Li Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2024. REASONER: an explainable recommendation dataset with comprehensive labeling ground truths. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA) *(NIPS '23)*. Curran Associates Inc., Red Hook, NY, USA, Article 638, 19 pages.

[2] MMEngine Contributors. 2022. MMEngine: OpenMMLab Foundational Library for Training Deep Learning Models. https://github.com/open-mmlab/mmengine. (2022).

[3] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-Observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (Atlanta, GA, USA) *(CIKM '22)*. 540–550. https://doi.org/10.1145/3511808.3557220

[4] Wei Guo, Fuzhen Zhuang, Xiao Zhang, Yiqi Tong, and Jin Dong. 2024. A comprehensive survey of federated transfer learning: challenges, methods and applications. *Front. Comput. Sci.* 18, 6 (July 2024), 34 pages. https://doi.org/10.1007/s11704-024-40065-x

[5] Yiheng Jiang, Yuanbo Xu, Yongjian Yang, Funing Yang, Pengyang Wang, and Hui Xiong. 2023. TriMLP: Revenge of a MLP-like Architecture in Sequential Recommendation. arXiv:2305.14675 [cs.LG]

[6] Diederik P Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations*.

[7] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. 2017. CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. arXiv:1709.02023 [cs.LG]

[8] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37. https://doi.org/10.1109/MC.2009.263

[9] Sébastien Lachapelle, Philippe Brouillard, Tristan Deleu, and Simon Lacoste-Julien. 2019. Gradient-Based Neural DAG Learning. *CoRR* abs/1906.02226 (2019). arXiv:1906.02226 http://arxiv.org/abs/1906.02226

[10] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. arXiv:1802.05814 [stat.ML]

[11] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 4114–4124. https://proceedings.mlr.press/v97/locatello19a.html

[12] Huishi Luo, Fuzhen Zhuang, Ruobing Xie, Hengshu Zhu, Deqing Wang, Zhulin An, and Yongjun Xu. 2024. A survey on causal inference for recommendation. *The Innovation* 5, 2 (2024).

[13] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. *Advances in neural information processing systems* 32 (2019).

[14] Christopher Maddison, Andriy Mnih, and Yee Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables.

[15] Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. [n. d.]. *Masked Gradient-Based Causal Structure Learning*. 424–432. https://doi.org/10.1137/1.9781611977172.48 arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611977172.48

[16] Judea Pearl. 2009. Causal inference in statistics: An overview. (2009).

[17] Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. 2014. Causal Discovery with Continuous Additive Noise Models. *Journal of Machine Learning Research* 15, 58 (2014), 2009–2053. http://jmlr.org/papers/v15/peters14a.html

[18] Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. 2022. Weakly Supervised Disentangled Generative Causal Representation Learning. *Journal of Machine Learning Research* 23, 241 (2022), 1–55. http://jmlr.org/papers/v23/21-0080.html

[19] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. 2020. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*. 528–536.

[20] Robert Tillman and Peter Spirtes. 2011. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 15)*, Geoffrey Gordon, David Dunson, and Miroslav Dudík (Eds.). PMLR, Fort Lauderdale, FL, USA, 3–15. https://proceedings.mlr.press/v15/tillman11a.html

[21] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[22] En Wang, Yiheng Jiang, Yuanbo Xu, Liang Wang, and Yongjian Yang. 2022. Spatial-temporal interval aware sequential POI recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2086–2098.

[23] Siyu Wang, Xiaocong Chen, Quan Z. Sheng, Yihong Zhang, and Lina Yao. 2023. Causal Disentangled Variational Auto-Encoder for Preference Understanding in Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Taipei, Taiwan) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1874–1878. https://doi.org/10.1145/3539618.3591961

[24] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 1717–1725. https://doi.org/10.1145/3447548.3467249

[25] Xin Wang, Hong Chen, Yuwei Zhou, Jianxin Ma, and Wenwu Zhu. 2023. Disentangled Representation Learning for Recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 1 (2023), 408–424. https://doi.org/10.1109/TPAMI.2022.3153112

[26] Zimu Wang, Yue He, Jiashuo Liu, Wenchao Zou, Philip S Yu, and Peng Cui. 2022. Invariant preference learning for general debiasing in recommendation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1969–1978.

[27] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*. 153–162.

[28] Hangtong Xu, Yuanbo Xu, and Yongjian Yang. 2025. Separating and Learning Latent Confounders to Enhancing User Preferences Modeling. In *Database Systems for Advanced Applications*, Makoto Onizuka, Jae-Gil Lee, Yongxin Tong, Chuan Xiao, Yoshiharu Ishikawa, Sihem Amer-Yahia, H. V. Jagadish, and Kejing Lu (Eds.). Springer Nature Singapore, Singapore, 67–82.

[29] Hangtong Xu, Yuanbo Xu, Yongjian Yang, Fuzhen Zhuang, and Hui Xiong. 2023. DPR: An Algorithm Mitigate Bias Accumulation in Recommendation feedback loops. *arXiv preprint arXiv:2311.05864* (2023).

[30] Shuyuan Xu, Jianchao Ji, Yunqi Li, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. Causal Inference for Recommendation: Foundations, Methods and Applications. arXiv:2301.04016 [cs.IR]

[31] Shuyuan Xu, Juntao Tan, Shelby Heinecke, Vena Jia Li, and Yongfeng Zhang. 2023. Deconfounded Causal Collaborative Filtering. *ACM Transactions on Recommender Systems* 1, 4 (oct 2023), 1–25. https://doi.org/10.1145/3606035

[32] Yuanbo Xu, En Wang, Yongjian Yang, and Yi Chang. 2022. A Unified Collaborative Representation Learning for Neural-Network Based Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2022), 5126–5139. https://doi.org/10.1109/TKDE.2021.3054782

[33] Yuanbo Xu, En Wang, Yongjian Yang, and Hui Xiong. 2024. GS-RS: A Generative Approach for Alleviating Cold Start and Filter Bubbles in Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* 36, 2 (2024), 668–681. https://doi.org/10.1109/TKDE.2023.3290140

[34] Yuanbo Xu, Fuzhen Zhuang, En Wang, Chaozhuo Li, and Jie Wu. 2025. Learning Without Missing-At-Random Prior Propensity-A Generative Approach for Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering* 37, 2 (2025), 754–765. https://doi.org/10.1109/TKDE.2024.3490593

[35] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9593–9602.

[36] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG Structure Learning with Graph Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 7154–7163. https://proceedings.mlr.press/v97/yu19a.html

[37] Qing Zhang, Xiaoying Zhang, Yang Liu, Hongning Wang, Min Gao, Jiheng Zhang, and Ruocheng Guo. 2023. Debiasing Recommendation by Learning Identifiable Latent Confounders. *arXiv preprint arXiv:2302.05052* (2023).

[38] Shengyu Zhang, Fuli Feng, Kun Kuang, Wenqiao Zhang, Zhou Zhao, Hongxia Yang, Tat-Seng Chua, and Fei Wu. 2023. Personalized Latent Structure Learning for Recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 8 (2023), 10285–10299. https://doi.org/10.1109/TPAMI.2023.3247563

[39] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation *(SIGIR '21)*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/3404835.3462875

[40] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. RecBole 2.0: Towards a More Up-to-Date Recommendation Library. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 4722–4726.

[41] Xun Zheng, Bryon Aragam, Pradeep Ravikumar, and Eric P. Xing. 2018. DAGs with NO TEARS: Continuous Optimization for Structure Learning. arXiv:1803.01422 [stat.ML]

[42] Xinyuan Zhu, Yang Zhang, Fuli Feng, Xun Yang, Dingxian Wang, and Xiangnan He. 2022. Mitigating Hidden Confounding Effects for Causal Recommendation. *ArXiv* abs/2205.07499 (2022). https://api.semanticscholar.org/CorpusID:248811475

[43] Yaochen Zhu, Jing Yi, Jiayi Xie, and Zhenzhong Chen. 2022. Deep Causal Reasoning for Recommendations. https://api.semanticscholar.org/CorpusID:245769824

# A PROOFS

## A.1 Proof of Evidence Lower Bound

$$\ln p_\theta(x_u) \geq E_{q(z,|x_u,sInfo_u,\mathbf{C})}\left[\ln p(x_u|z,sInfo_u,\mathbf{C})\right] - D_{KL}(q(z|x_u,sInfo_u,\mathbf{C})\|p(z|sInfo_u,\mathbf{C})).$$

We give the proofs as follows:

Proof. The mixed preference $z$ is obtained from the conditional distribution $p(z|sInfo_u, \mathbf{C})$, and $sInfo_u$ and $\mathbf{C}$ are independent of each other, we need to adjust the derivation of the VAE's ELBO to account for this new conditional distribution.

We can write the conditional distribution $p(z|sInfo_u, \mathbf{C})$ and incorporate these conditions into the ELBO derivation. The marginal log-likelihood of the data can be expressed as:

$$\log p(x_u|sInfo_u, \mathbf{C}) = \log \int p(x_u, z|sInfo_u, \mathbf{C})\, dz.$$

We introduce the variational distribution $q(z|x_u, sInfo_u, \mathbf{C})$ to approximate the posterior distribution $p(z|x_u, sInfo_u, \mathbf{C})$, and use Jensen's inequality to derive the ELBO:

$$\log p(x_u|sInfo_u, \mathbf{C}) = \log \int q(z|x_u, sInfo_u, \mathbf{C}) \frac{p(x_u, z|sInfo_u, \mathbf{C})}{q(z|x_u, sInfo_u, \mathbf{C})}\, dz.$$

$$\log p(x_u|sInfo_u, \mathbf{C}) = \log \mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \frac{p(x_u, z|sInfo_u, \mathbf{C})}{q(z|x_u, sInfo_u, \mathbf{C})} \right] \geq \mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log \frac{p(x_u, z|sInfo_u, \mathbf{C})}{q(z|x_u, sInfo_u, \mathbf{C})} \right].$$

The expectation on the right is the ELBO:

$$\mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log \frac{p(x_u, z|sInfo_u, \mathbf{C})}{q(z|x_u, sInfo_u, \mathbf{C})} \right] = \mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log p(x_u, z|sInfo_u, \mathbf{C}) - \log q(z|x_u, sInfo_u, \mathbf{C}) \right].$$

We decompose the joint distribution $p(x_u, z|sInfo_u, \mathbf{C})$ as follows:

$$p(x_u, z|sInfo_u, \mathbf{C}) = p(x_u|z, sInfo_u, \mathbf{C}) p(z|sInfo_u, \mathbf{C}).$$

Thus, the ELBO can be further decomposed as:

$$\begin{aligned} &\mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log p(x_u, z|sInfo_u, \mathbf{C}) - \log q(z|x_u, sInfo_u, \mathbf{C}) \right] \\ =&\mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log p(x_u|z, sInfo_u, \mathbf{C}) + \log p(z|sInfo_u, \mathbf{C}) - \log q(z|x_u, sInfo_u, \mathbf{C}) \right]. \end{aligned}$$

We can rewrite it as the sum of two parts:

$$\mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log p(x_u|z, sInfo_u, \mathbf{C}) \right] - D_{KL}(q(z|x_u, sInfo_u, \mathbf{C}) \| p(z|sInfo_u, \mathbf{C})),$$

where $D_{KL}(q(z|x_u, sInfo_u, \mathbf{C}) \| p(z|sInfo_u, \mathbf{C}))$ is the Kullback-Leibler divergence between the variational distribution $q(z|x_u, sInfo_u, \mathbf{C})$ and the conditional prior distribution $p(z|sInfo_u, \mathbf{C})$.

In summary, when $z$ is obtained from the conditional distribution $p(z|sInfo_u, \mathbf{C})$, and $sInfo_u$ and $\mathbf{C}$ are independent, the ELBO of the VAE is:

$$\text{ELBO} = \mathbb{E}_{q(z|x_u, sInfo_u, \mathbf{C})} \left[ \log p(x_u|z, sInfo_u, \mathbf{C}) \right] - D_{KL}(q(z|x_u, sInfo_u, \mathbf{C}) \| p(z|sInfo_u, \mathbf{C})).$$

□

### A.2 Synthetic Dataset Details.

For synthetic data experiments, the number of user samples is 300, and for each user sample, 500 items. We assume that users are influenced by four different categories of confounders, and the causal relationships of these four confounders can be generated using Eq 2. Specifically, we consider the following causal structural model:

$$\begin{aligned} n_1 &= \mathcal{N}(\lambda_1, \beta_1), \\ n_2 &= \mathcal{N}(\lambda_2, \beta_2), \\ n_3 &= \mathcal{N}(\lambda_3, \beta_3), \\ n_4 &= \mathcal{N}(\lambda_4, \beta_4), \\ c_1 &= n_1, \\ c_2 &= w_2(u)(c_1) + n_2, \\ c_3 &= w_3(u)(c_2) + n_3, \\ c_4 &= w_4(u)(c_2) + n_4, \end{aligned} \tag{24}$$

where $\lambda_i$ and $\beta_i$ are the mean and variance of the Gaussian distribution, we sample the $\lambda_i$ and $\beta_i$ from the uniform distribution $[-3, 3]$ and $[0.01, 4]$, respectively. We sample the user-specific weight $w_i$ from Poisson distribution with the give $u$:

$$u = \mathcal{N}(0, 1). \tag{25}$$

Finally, given the set of confounders and users, we can generate the final observed value § through a non-linear function $g(\cdot)$:

$$\mathcal{X} = g(c_1, c_2, c_3, c_4, u). \tag{26}$$

In our experiments, we used a two-layer MLP for the blending generation.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009