



# Causal Disentangled Recommendation Against User Preference Shifts

WENJIE WANG, National University of Singapore, Singapore

XINYU LIN, National University of Singapore, Singapore

LIUHUI WANG, Peking University, China

FULI FENG\*, University of Science and Technology of China, China

YUNSHAN MA, National University of Singapore, Singapore

TAT-SENG CHUA, National University of Singapore, Singapore

Recommender systems easily face the issue of user preference shifts. User representations will become out-of-date and lead to inappropriate recommendations if user preference has shifted over time. To solve the issue, existing work focuses on learning robust representations or predicting the shifting pattern. There lacks a comprehensive view to discover the underlying reasons for user preference shifts. To understand the preference shift, we abstract a causal graph to describe the generation procedure of user interaction sequences. Assuming user preference is stable within a short period, we abstract the interaction sequence as a set of chronological environments. From the causal graph, we find that the changes of some unobserved factors (e.g., becoming pregnant) cause preference shifts between environments. Besides, the fine-grained user preference over item categories sparsely affects the interactions with different items. Inspired by the causal graph, our key considerations to handle preference shifts lie in modeling the interaction generation procedure by: 1) capturing the preference shifts across environments for accurate preference prediction, and 2) disentangling the sparse influence from user preference to interactions for accurate effect estimation of preference. To this end, we propose a Causal Disentangled Recommendation (CDR) framework, which captures preference shifts via a temporal variational autoencoder and learns the sparse influence from multiple environments. Specifically, an encoder is adopted to infer the unobserved factors from user interactions while a decoder is to model the interaction generation process. Besides, we introduce two learnable matrices to disentangle the sparse influence from user preference to interactions. Lastly, we devise a multi-objective loss to optimize CDR. Extensive experiments on three datasets show the superiority of CDR in enhancing the generalization ability under user preference shifts.

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

Additional Key Words and Phrases: Causal Disentangled Recommendation, Preference Shifts, Generalizable Recommendation, Out-of-Distribution Generalization

\*Corresponding author: Fuli Feng (fulifeng93@gmail.com). This work is supported by the Defence Science and Technology Agency and the National Natural Science Foundation of China (62272437).

Authors' addresses: Wenjie Wang, wenjiewang96@gmail.com, National University of Singapore, Singapore; Xinyu Lin, xylin1028@gmail.com, National University of Singapore, Singapore; Liuhui Wang, wangliuhui0401@pku.edu.cn, Peking University, China; Fuli Feng, fulifeng93@gmail.com, University of Science and Technology of China, China; Yunshan Ma, yunshan.ma@u.nus.edu, National University of Singapore, Singapore; Tat-Seng Chua, dcscts@nus.edu.sg, National University of Singapore, Singapore.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

1046-8188/2023/4-ART \$15.00

<https://doi.org/10.1145/3593022>

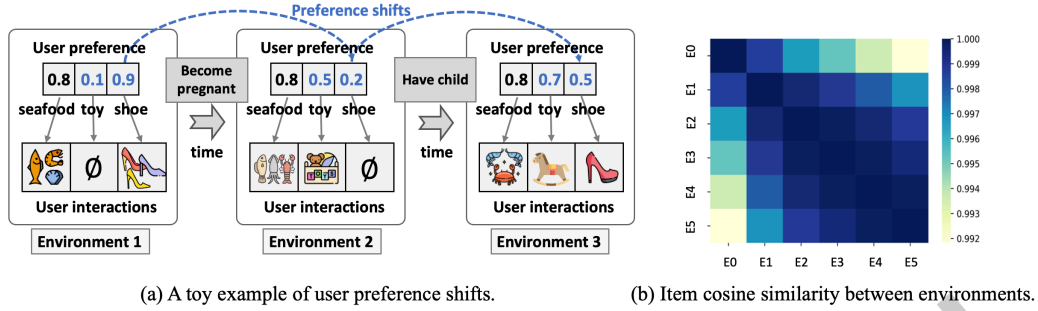


Fig. 1. (a) is a toy example to illustrate preference shifts. We assume that user preference is stable within a short period, thus treating a short time period as an environment. Some user preference is stable across environments, *e.g.*, the preference over seafood; while partial preference is shifting due to the changes of user features (*e.g.*, becoming pregnant). (b) shows the cosine similarity of item categories interacted by users in different environments of Amazon Book. The environments are chronologically split, starting from E0 to E5. It shows that item similarity decreases over time, revealing the interaction distribution shifts.

## 1 INTRODUCTION

Recommender models typically learn user preference representations from historical interactions (*e.g.*, clicks and ratings) [10]. However, most recommender models assume the training and testing interactions are Independent and Identically Distributed (IID), which is infeasible in real-world applications. As shown in Figure 1, there are user preference shifts across different environments [57, 75] where each environment denotes a short time period. Some user features and environmental factors will change over time such as becoming pregnant, causing the shifts of user preference and interaction distributions in Out-of-Distribution (OOD) environments. Such drifts can frustrate the recommender models trained over historical interactions [57]. Consequently, the inferior performance of recommender models will degrade user experience and reduce user activities, hurting the health of the whole recommender system. Therefore, it is essential to capture user preference shifts and pursue generalizable recommendation.

Existing work mainly handles preference shifts from two perspectives:

- *Robust models against preference shifts.* The most representative approach is disentangled recommendation [31, 32], which disentangles several independent representations to represent different user preference. Disentangled representations are less sensitive to preference shifts [31] since only partial representations are shifted while most are robust in the OOD environment. Nevertheless, existing work usually ignores the temporal shifting patterns of user preference across environments, limiting the generalization ability of recommender models.
- *Sequential models to predict the shifts.* Sequential recommendation [67, 76] recognizes the preference shifts by modeling the temporal patterns within the interaction sequence. However, these sequential models typically overlook the sparse influence of user preference on interactions: partial preference shifts between environments only affect a small portion of user interactions. Ignoring such sparse influence may harm the interaction predictions over extensive irrelevant items, resulting in many inappropriate recommendations in OOD environments.

There lacks a comprehensive view to reveal the underlying factors regarding user preference shifts. As such, we resort to causal language to inspect the causal relations behind the generation procedure of user interaction sequences. As shown in Figure 2, the interaction sequence is divided into multiple short time periods, where each period is viewed as an environment.  $E_t$  represents the unobserved user features (*e.g.*, pregnancy) or environmental

factors<sup>1</sup> (e.g., hot events) in the environment  $t$ , which affect the user preference  $Z_t$  and interactions  $X_t$ . Within each environment, user preference is relatively stable; however, some changes from  $E_{t-1}$  to  $E_t$  (e.g., becoming pregnant) will shift user preference from  $Z_{t-1}$  to  $Z_t$ . As to the effect of  $Z_t$  on  $X_t$ , various category-level preference in  $Z_t$  sparsely affects the interactions in  $X_t$  as shown in the right part of Figure 2. For instance, the preference over the category “Toy” influences the interactions with toy products. Due to the sparse influence, partial preference shifts from  $Z_{t-1}$  to  $Z_t$  only affect some interactions.

According to the causal relations, the key of handling user preference shifts lies in simultaneously 1) capturing the temporal shifts across environments ( $Z_{t-1} \rightarrow Z_t$ ), i.e., accurate preference prediction, and 2) disentangling the sparse influence from user preference to the interactions ( $Z_t \rightarrow X_t$ ), i.e., accurate effect estimation of preference. Disentangling such influence is essentially discovering the causal structure from  $Z_t$  to  $X_t$  [11, 34]. However, it is non-trivial to model the temporal preference shifts and the sparse influence due to the following challenges: 1) the changes of  $E_t$  between environments are usually unobserved, hindering the accurate preference prediction; and 2) the causal structure from hidden user preference to interactions also lacks supervision, which requires us to find additional signals for the structure discovery between  $Z_t$  and  $X_t$ .

To this end, we propose a Causal Disentangled Recommendation (CDR) framework, which models the interaction generation procedure in Figure 2. Specifically, 1) to estimate the unobserved  $E_t$ , CDR introduces a temporal Variational AutoEncoder (VAE), where an encoder uses variational inference to infer unobserved  $E_t$  from observed interactions  $X_t$ . Besides, a decoder is to estimate the effect of  $E_t$  on  $X_t$  via  $Z_t$ , where  $Z_t$  is iteratively updated to model the temporal shifts across environments. 2) Furthermore, we introduce two learnable matrices to formulate the causal structure from  $Z_t$  to  $X_t$ . The two matrices disentangle the representations of  $Z_t$  into category-level preference, which then sparsely affects the interactions in corresponding categories. Due to lacking supervision, we propose to learn the two matrices from multiple environments, where the distribution shifts shed light on the sparse structure learning between  $Z_t$  and  $X_t$  [27, 44]. In particular, we utilize the variance regularization to balance the predictions across environments and adopt the sparsity regularization to control the sparsity of the structure. Note that the VAE ignores the temporal information and fairly considers each interaction within each environment, and thus it captures the invariant preference in a short period. As such, during training and inference, we can flexibly adjust the division of environments to balance the modeling of invariant preference within environments and shifted preference between environments. Extensive experiments on three real-world datasets validate the effectiveness of CDR in capturing preference shifts and achieving superior performance in the OOD environments. We release the code and data at <https://github.com/Linxyhaha/CDR>.

The main contributions of this work are threefold:

- We retrospect user preference shifts across multiple environments from a causal view and inspect the underlying causal relations via a causal graph.
- We propose a CDR framework, which captures the preference shifts between environments via a temporal VAE and learns a sparse structure between user preference and interactions for the robust interaction prediction.
- Empirical results on three public datasets demonstrate the superiority of CDR over the baselines *w.r.t.* the OOD generalization ability under preference shifts.

## 2 METHOD

We present the causal mechanism and task formulation of recommendation with consideration of preference shifts in Section 2.1. Thereafter, we detail the proposed CDR framework in Section 2.2.

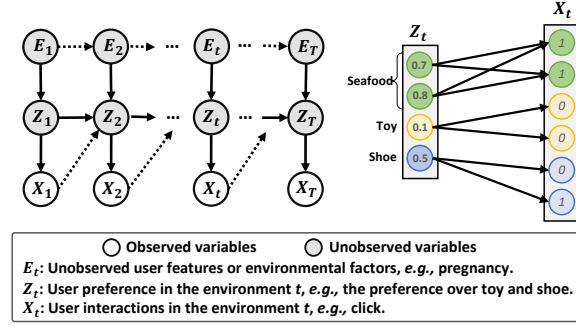


Fig. 2. Causal graph behind the interaction generation procedure with multiple environments. We assume that the observed user preference is affected by some hidden user features, and various user preference sparsely influences the interactions with different items.

## 2.1 Recommendation with Preference Shifts

Existing recommender models usually rely on the IID interaction distributions from training to testing stages. Without considering user preference shifts over time, these recommender models will encounter significant a performance drop in OOD environments. To improve the generalization ability, we build recommender models by considering the preference shifts. In this subsection, we first scrutinize the causal relations regarding user preference shifts, and then formulate the task of generalizable recommendation to evaluate the generalization ability under preference shifts.

**2.1.1 Causal View of Preference Shifts.** We present the causal relations in Figure 2 and explain its rationality as follows:

- $E_t$  denotes unobserved user features (e.g., pregnancy) or environmental factors (e.g., hot events) in the environment  $t$ ;  $Z_t$  and  $X_t$  represent the user preference and interactions, respectively. Because of the privacy restriction [52], we seldom utilize user features for recommendation, and thus we ignore the modeling of observed user features in Figure 2, which can be easily incorporated as the input of the CDR framework if necessary [57].
- $E_t \rightarrow Z_t$ : user features and various environmental factors affect user preference.
- $Z_t \rightarrow X_t$ : user interactions are determined by current user preference. In particular,  $Z_t$  covers the preference over multiple item categories (e.g., seafood and toy). Some factors in  $Z_t$  may represent the preference over an item category (e.g., seafood), which sparsely affects a category of interactions as shown in Figure 2.
- $Z_{t-1} \rightarrow Z_t$ : the user preference  $Z_t$  in the environment  $t$  is updated from previous  $Z_{t-1}$ , which exhibits the preference shifts over time. From the causal graph, we find that various factors in  $E_t$  can affect user preference  $Z_t$  and cause the preference shifts  $Z_{t-1} \rightarrow Z_t$ , leading to the variation of user interaction distributions. Besides, the preference shifts between environments only influence partial interactions due to the sparse influence from  $Z_t$  to  $X_t$ .
- $E_{t-1} \leftrightarrow E_t$  and  $X_{t-1} \leftrightarrow Z_t$ :  $E_{t-1}$  might affect  $E_t$  because user features might have conditional relations, e.g., pregnancy  $\rightarrow$  having child. Besides, user preference  $Z_t$  can be influenced by previous interactions  $X_{t-1}$ . However, these conditional relations are not easy to be inferred from pure interactions, and the effects of these conditional relations on  $Z_t$  and  $X_t$  are relatively weaker than  $(E_t, Z_{t-1}) \rightarrow Z_t$  and  $Z_t \rightarrow X_t$ . As such, we omit the modeling of  $E_{t-1} \leftrightarrow E_t$  and  $X_{t-1} \leftrightarrow Z_t$  in this work to pursue a simple model with fewer parameters. Empirical evidence in Section 3.3.3 also validates the superiority of our choice.

<sup>1</sup>Recommender models can observe few factors affecting preference due to privacy restriction and technical challenges [52].

**2.1.2 Task Formulation.** To evaluate the generalization ability under preference shifts, we formulate the task of generalizable recommendation. Formally, we utilize  $u \in \{1, 2, \dots, U\}$ ,  $i \in \{1, 2, \dots, I\}$ , and  $t \in \{1, 2, \dots, T\}$  to index the user, item, and environment, respectively. The interactions of user  $u$  in  $T$  environments are denoted as  $\mathbf{x}_{1:T}$ <sup>2</sup>, where  $\mathbf{x}_t \in \{0, 1\}^I$  is a multi-hot vector, and  $x_{t,i}$  implies that user  $u$  likes item  $i$  ( $x_{t,i} = 1$ ) or not ( $x_{t,i} = 0$ ). Generally, given the observed  $\mathbf{x}_{1:T}$  of user  $u$ , **generalizable recommendation** aims to capture the hidden preference shifts in  $\mathbf{z}_{1:T}$  and estimates the latest user preference  $\mathbf{z}_T$ .

- **Environment division.** We can divide the environments by time, for instance, equally splitting the user interaction sequence into  $T$  pieces, or clustering adjacent interactions according to the time interval. In this work, we choose the first one to simplify the data pre-processing.
- **Inference for future environments.** To evaluate the generalization ability of recommender models, we can utilize the interactions in the environment  $T + 1$  for testing. To infer the interaction probability in this unknown environment, we consider three strategies: 1) using the latest user preference  $\mathbf{z}_T$  for prediction; 2) uniformly averaging the predictions in  $T$  training environments; 3) considering the average user features  $\mathbf{e}_{T+1} = \frac{1}{T} \sum_{t=1}^T \mathbf{e}_t$  and  $\mathbf{z}_T$  to predict  $\mathbf{z}_{T+1}$ , and then using  $\mathbf{z}_{T+1}$  for interaction prediction. Because the testing environment is unknown, these inference strategies inevitably make some assumptions. The first strategy requires the small preference shifts from environment  $T$  to  $T + 1$ . Meanwhile, the second and third strategies need the average over  $T$  training environments, losing some temporal shifting patterns. In practice, we set the first strategy as the default due to its better performance on real-world datasets (refer to Section 3.3.4).
- **Difference from sequential recommendation.** The main difference between generalizable and sequential recommendations is that generalizable recommendation emphasizes the preference shifts across environments and the invariant preference within an environment. Moreover, generalizable recommendation focuses on the predictions of multiple interactions in an OOD environment, which differs from the next-item prediction in sequential recommendation.

## 2.2 CDR Framework

In this subsection, we present the CDR framework to model the interaction generation procedure under multiple environments. In particular, we utilize a novel temporal VAE to capture the preference shifts ( $\mathbf{z}_{t-1} \rightarrow \mathbf{z}_t$ ) and conduct sparse structure learning to disentangle the sparse influence from user preference to interactions ( $\mathbf{z}_t \rightarrow \mathbf{x}_t$ ).

We construct the recommender model by following the causal relations in Figure 2. Specifically, for each user  $u$  in the environment  $t$ , we first sample a  $K$ -dimensional latent representation  $\mathbf{e}_t$  from the standard Gaussian prior  $\mathcal{N}(0, \mathbf{I}_K)$  [25, 71], where the covariance  $\mathbf{I}_K$  is an identity matrix. We then obtain user preference  $\mathbf{z}_t \in \mathbb{R}^H$  based on  $\mathbf{e}_t$  and the previous  $\mathbf{z}_{t-1}$ . Thereafter,  $\mathbf{z}_t$  is used to predict the interaction probability over  $I$  items and the historical interactions  $\mathbf{x}_t \in \mathbb{R}^I$  are assumed to be drawn from the interaction probability distribution. In this work, we assume that  $\mathbf{z}_t$  and  $\mathbf{x}_t$  follow the factorized Gaussian and multinomial priors due to their superiority shown in previous work [25, 31]. Formally,

$$\begin{cases} \mathbf{e}_t \sim \mathcal{N}(0, \mathbf{I}_K), \\ \mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{\theta_1}(\mathbf{e}_t, \mathbf{z}_{t-1}), \text{diag}\{\boldsymbol{\sigma}_{\theta_1}^2(\mathbf{e}_t, \mathbf{z}_{t-1})\}), \\ \mathbf{x}_t \sim \text{Mult}(N_t, \pi(f_{\theta_2}(\mathbf{z}_t))). \end{cases} \quad (1)$$

Specifically, we explain the generative process in Eq. (1), which is consistent with the causal relations in Figure 2:

- $(\mathbf{e}_t, \mathbf{z}_{t-1}) \rightarrow \mathbf{z}_t$ :  $\boldsymbol{\mu}_{\theta_1}(\mathbf{e}_t, \mathbf{z}_{t-1})$  and  $\boldsymbol{\sigma}_{\theta_1}^2(\mathbf{e}_t, \mathbf{z}_{t-1})$  denote the *mean* and *diagonal covariance* of the Gaussian distribution of  $\mathbf{z}_t$ , which are estimated from  $\mathbf{e}_t$  and  $\mathbf{z}_{t-1}$  via a network  $f_{\theta_1}(\cdot)$ .

<sup>2</sup>For notation brevity, we omit the subscript  $u$  for  $\mathbf{x}_{1:T}$  and  $\mathbf{z}_{1:T}$  of user  $u$ .

- $\mathbf{z}_t \rightarrow \mathbf{x}_t$ :  $\mathbf{x}_t$  is sampled from a multinomial distribution affected by  $\mathbf{z}_t$ , where  $N_t = \sum_{i=1}^I x_{t,i}$  represents the interaction number of user  $u$  in the environment  $t$ ,  $\pi(\cdot)$  is the *softmax* function, and the network  $f_{\theta_2}(\mathbf{z}_t)$  transforms  $\mathbf{z}_t$  to produce the interaction probability over  $I$  items.

To train the recommender model, we aim to optimize the parameters  $\{\theta_1, \theta_2\}$  by maximizing the generative probability of observed user interactions  $\mathbf{x}_{1:T}$  in  $T$  environments. Formally, following [5], we can factorize the joint distribution  $p(\mathbf{x}_{1:T})$  and maximize the log-likelihood as follows:

$$\begin{aligned} \log p(\mathbf{x}_{1:T}) &= \log \int p(\mathbf{x}_{1:T}|\mathbf{e}_{1:T})p(\mathbf{e}_{1:T})d\mathbf{e}_{1:T} \\ &= \log \int \prod_{t=1}^T p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})p(\mathbf{e}_{1:T})d\mathbf{e}_{1:T}, \end{aligned} \quad (2)$$

where  $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})$  aligns with the generation procedure in Eq. (1) and we will further factorize it with  $\mathbf{z}_{1:t}$  in the decoding process *i.e.*, Eq. (5). Nevertheless, maximizing Eq. (2) is intractable because it involves the integral over unobserved  $\mathbf{e}_{1:T}$ . To solve the problem, we embrace *variational inference* [25] to approximate  $\log p(\mathbf{x}_{1:T})$  by using a variational distribution  $q(\mathbf{e}_{1:T}|\cdot)$ . Formally,

$$\log p(\mathbf{x}_{1:T}) = \log \int \prod_{t=1}^T p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})p(\mathbf{e}_{1:T}) \frac{q(\mathbf{e}_{1:T}|\cdot)}{q(\mathbf{e}_{1:T}|\cdot)} d\mathbf{e}_{1:T} \quad (3a)$$

$$\geq \mathbb{E}_{q(\mathbf{e}_{1:T}|\cdot)} \left[ \log \frac{\prod_{t=1}^T p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})p(\mathbf{e}_{1:T})}{q(\mathbf{e}_{1:T}|\cdot)} \right] \quad (\text{ELBO}) \quad (3b)$$

$$= \mathbb{E}_{q(\mathbf{e}_{1:T}|\cdot)} \left[ \sum_{t=1}^T (\log p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t}) - \text{KL}[q(\mathbf{e}_t|\cdot)||p(\mathbf{e}_t)]) \right], \quad (3c)$$

where variational inference introduces the Evidence Lower BOund (ELBO) of Eq. (3a) by using  $q(\mathbf{e}_{1:T}|\cdot) = \prod_{t=1}^T q(\mathbf{e}_t|\cdot)$ . Meanwhile, the first term in Eq. (3c) represents the probability of collecting observed  $\mathbf{x}_t$  conditioned on  $\mathbf{x}_{1:t-1}$  and  $\mathbf{e}_{1:t}$  while the second term denotes the Kullback-Leibler (KL) divergence between the variational distribution  $q(\mathbf{e}_t|\cdot)$  and the prior of  $\mathbf{e}_t$ . By maximizing the ELBO in Eq. (3c), we are able to increase the log-likelihood  $\log p(\mathbf{x}_{1:T})$ . Note that we avoid factorizing  $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})$  with  $\mathbf{z}_{1:t}$  in Eq. (3), and then we do not estimate the distribution of unobserved  $\mathbf{z}_t$  and  $\mathbf{e}_t$  simultaneously by variational inference. This is because we choose the alternative Monte Carlo (MC) sampling to efficiently approximate the posterior distribution of  $\mathbf{z}_t$  [4]. MC sampling constructs a random sampling of  $\mathbf{z}_t$  (*i.e.*, draw samples from  $p(\mathbf{z}_t|\mathbf{e}_t, \mathbf{z}_{t-1})$ ) to estimate its distribution, which avoids unnecessary prior hypothesis over the mean and covariance of  $\mathbf{z}_t$  (see Eq. (5)).

So far, the key of calculating the ELBO in Eq. (3c) lies in estimating  $q(\mathbf{e}_t|\cdot)$  and  $p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \mathbf{e}_{1:t})$ , which can be obtained by the encoder and decoder networks, respectively. We present the intuitive illustration of CDR with the encoder and decoder networks in Figure 3.

**2.2.1 Encoder Network.** To estimate  $q(\mathbf{e}_t|\cdot)$ , we incorporate an encoder network  $g_\phi(\cdot)$ , which predicts  $\mathbf{e}_t$  by the user interaction  $\mathbf{x}_t$ . The underlying motivation is that unobserved factors (*e.g.*, income) can be inferred from users' behaviors (*e.g.*, purchasing expensive products). In particular,

$$q(\mathbf{e}_t|\cdot) = q(\mathbf{e}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{e}_t; \boldsymbol{\mu}_\phi(\mathbf{x}_t), \text{diag}\{\boldsymbol{\sigma}_\phi^2(\mathbf{x}_t)\}), \quad (4)$$

where  $\boldsymbol{\mu}_\phi(\mathbf{x}_t)$  and  $\boldsymbol{\sigma}_\phi^2(\mathbf{x}_t)$  denote the mean and diagonal covariance of  $\mathbf{e}_t$ , respectively. They are estimated by the encoder network  $g_\phi(\cdot)$  parameterized by  $\phi$ . Formally, we have  $g_\phi(\mathbf{x}_t) = [\boldsymbol{\mu}_\phi(\mathbf{x}_t), \boldsymbol{\sigma}_\phi(\mathbf{x}_t)] \in \mathbb{R}^{2K}$ . In this work,

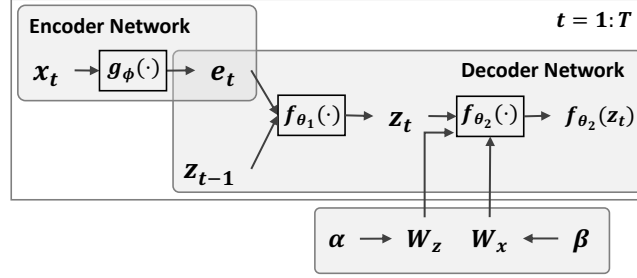


Fig. 3. Illustration of the CDR framework, where the encoder network predicts the hidden user features  $e_t$  in the environment  $t$ , and then the decoder reconstructs the interaction generation procedure from  $e_t$  and  $z_{t-1}$  to the interaction probability  $f_{\theta_2}(z_t)$ . The entire encoder-decoder process repeats  $T$  times while  $W_z$  and  $W_x$  are shared across  $T$  environments.

we instantiate  $g_\phi(\cdot)$  by a Multi-Layer Perceptron (MLP), which outputs the Gaussian parameters of  $e_t$ . Note that the encoder ignores the temporal interaction sequence in  $x_t$  and fairly encodes every interaction.

**2.2.2 Decoder Network.** We factorize  $p(x_t|x_{1:t-1}, e_{1:t})$  by following the causal relations in the interaction generation process:

$$p(x_t|x_{1:t-1}, e_{1:t}) = \int p(x_t|z_t) \prod_{a=1}^t p(z_a|z_{a-1}, e_a) dz_{1:t}, \quad (5)$$

where  $p(z_a|z_{a-1}, e_a)$  denotes the probability distribution of the user preference  $z_a$  in the environment  $a$ ; and when  $a = 1$ ,  $z_0$  is set as the constant vector  $\mathbf{0}$ . Besides, to approximate the distribution of  $z_t$ , we use MC sampling [4] to draw samples from  $p(z_t|z_{t-1}, e_t)$ . Then we can calculate  $p(x_t|x_{1:t-1}, e_{1:t})$  based on  $p(x_t|z_t)$  while marginalizing over  $z_{1:t}$  via the samples from MC sampling. To iteratively obtain  $p(z_t|z_{t-1}, e_t)$ , we adopt an MLP model  $f_{\theta_1}(\cdot)$  to output the  $\mu_{\theta_1}(\cdot)$  and  $\sigma_{\theta_1}(\cdot)$  of  $z_t$ . Formally, we have  $f_{\theta_1}(z_{t-1}, e_t) = [\mu_{\theta_1}(z_{t-1}, e_t), \sigma_{\theta_1}(z_{t-1}, e_t)] \in \mathbb{R}^{2H}$ . Thereafter, the remaining challenge is estimating  $p(x_t|z_t)$  in Eq. (5).

- **Sparse structure learning.** To estimate  $p(x_t|z_t)$ , we incorporate  $f_{\theta_2}(\cdot)$  to transform  $z_t$  into the interaction probability over  $I$  items. However, to align with the causal relations in Figure 2 and learn the sparse influence from user preference to interactions, we do not simply use an MLP model for the implementation of  $f_{\theta_2}(\cdot)$ . We instead resort to *sparse structure learning* in multiple environments, which aims to discover a sparse structure from user preference representations to interactions and requires the structure is robust across all the environments with distribution shifts. Consequently, 1) the sparse structure learned from multiple environments instead of one environment will encode the robust relations between user representations and interactions, which are likely to be reliable in future environments with preference shifts; and 2) if partial user preference has shifted, only a subset of interactions are affected due to the sparse structure. Such characteristics will improve the generalization ability of CDR under preference shifts.

Following [31], the user preference representation  $z_t \in \mathbb{R}^H$  can cover the preference over multiple item categories, and we disentangle  $z_t$  into several category-specific preference representations. Specifically, to implement the sparse structure, we introduce a matrix  $W_z \in \mathbb{R}^{H \times C}$  to factorize the user representation into the preference over  $C$  item categories. In particular,  $W_z[h, c] \in W_z$  denotes the probability of the  $h$ -th factor in  $z_t$  belonging to the preference over the  $c$ -th category. Correspondingly, we leverage a matrix  $W_x \in \mathbb{R}^{I \times C}$  to classify items into  $C$  categories. Inspired by [27, 70], we draw  $W_z$  and  $W_x$  from the clipped Gaussian distributions parameterized by  $\alpha \in \mathbb{R}^{H \times C}$  and  $\beta \in \mathbb{R}^{I \times C}$ , respectively. Formally, for each  $W_z[h, c] \in W_z$  and  $W_x[i, c] \in W_x$ , we

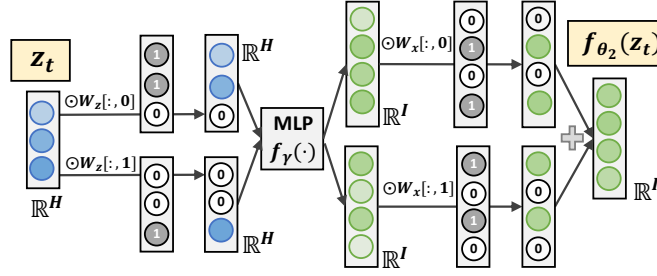


Fig. 4. Illustration of the calculation of  $f_{\theta_2}(z_t)$  in Eq. (7). Similar to masking mechanisms,  $W_z$  and  $W_x$  disentangle the user preference representations, leading to sparse connection from user preference to interactions. Note that we simplify  $W_z$  and  $W_x$  as discrete matrices with  $\{0, 1\}$  for better understanding.

have

$$\begin{cases} W_z[h, c] = \min(\max(\alpha_{h,c} + \epsilon, 0), 1), \\ W_x[i, c] = \min(\max(\beta_{i,c} + \epsilon, 0), 1), \end{cases} \quad (6)$$

where the noise  $\epsilon$  is drawn from  $\mathcal{N}(0, \sigma_\epsilon^2)$ . We clip the values of  $W_z$  and  $W_x$  into  $[0, 1]$  to ensure a valid range for the probabilities. Besides, to encourage each factor or item belonging to one category, we add a softmax function at the dimension of  $C$  categories in  $W_z$  and  $W_x$ . As illustrated in Figure 4, we then implement  $f_{\theta_2}(\cdot)$  to estimate the parameters of  $x_t$  in Eq. (1) by

$$f_{\theta_2}(z_t) = \sum_{c=1}^C W_x[:, c] \odot f_{\gamma}(W_z[:, c] \odot z_t), \quad (7)$$

where  $\theta_2 = \{\alpha, \beta, \gamma\}$ ,  $\odot$  denotes the element-wise multiplication, and  $f_{\gamma}(\cdot)$  can be any function transforming  $z_t$  to the interaction probability distribution over  $I$  items. Following [25], we implement  $f_{\gamma}(\cdot)$  by an MLP model.

• **Likelihood estimation.** As shown in Figure 3, given the user interactions  $x_{1:T}$  of user  $u$ , we feed them into the encoder network to sample  $e_{1:T}$ , and then iteratively pass  $e_{1:T}$  to the decoder network to obtain the parameters of the multinomial distribution for  $x_{1:T}$  (i.e.,  $f_{\theta_2}(z_{1:T})$ ). Thereafter, the log-likelihood  $\log p(x_t|z_t)$  can be calculated by

$$\log p(x_t|z_t) \stackrel{c}{=} \sum_{i=1}^I x_{t,i} \log \pi_i(f_{\theta_2}(z_t)), \quad (8)$$

where  $x_{t,i} \in \{0, 1\}$  indicates whether user  $u$  has interacted with item  $i$  in the environment  $t$  or not. Besides, the softmax function  $\pi(\cdot)$  is used to normalize  $f_{\theta_2}(\cdot)$  and  $\pi_i(f_{\theta_2}(\cdot))$  denotes the normalized prediction score for item  $i$ . Intuitively, the log-likelihood  $p(x_t|z_t)$  estimates the probability of drawing observed  $x_t$  from the multinomial distribution by sampling  $N_t$  times, where  $N_t$  is the interaction number of user  $u$  in the environment  $t$ .

**2.2.3 CDR Optimization.** We maximize the ELBO to increase the log-likelihood in Eq. (2) by optimizing the parameters (i.e.,  $\phi$  and  $\theta = \{\theta_1, \theta_2\}$ ) in CDR. The parameters are updated by stochastic gradient descent. However, we conduct the sampling of  $e_t$  in Eq. (4) and  $z_t$  in Eq. (5), which prevents the back-propagation of gradients. To solve this problem, we utilize the *reparameterization trick* [19, 25]. Besides, we leverage the *KL annealing trick* [25] to control the effect of the KL divergence, which introduces an additional hyper-parameter  $\lambda_1$  into Eq. (3c). To summarize, the optimization objective for user  $u$  is to minimize the following loss:

$$\mathcal{L}^u = -\mathbb{E}_{q_{\phi}(e_{1:T}|\cdot)} \left[ \sum_{t=1}^T (\log p(x_t|x_{1:t-1}, e_{1:t}) - \lambda_1 \text{KL}[q(e_t|x_t)||p(e_t)]) \right],$$



which becomes negative timestamp-wise ELBO [5] over the  $T$  environments.

- **Sparsity and variance regularization.** In addition to the ELBO objective, we additionally consider two regularization terms: 1) the sparsity of  $\mathbf{W}_z$  and  $\mathbf{W}_x$ , and 2) the variance of the gradients across  $T$  environments. In Eq. (7), we expect that the structure implemented by  $\mathbf{W}_z$  and  $\mathbf{W}_x$  is sparse because the sparse connection between user representations and interactions is more robust under preference shifts. Therefore, we introduce an  $L_0$  regularization term  $\|\mathbf{W}_z\|_0 + \|\mathbf{W}_x\|_0$  to restrict the number of non-zero values.

As to the variance regularization, it can facilitate the sparse structure learning across multiple environments. Training over multiple environments easily leads to imbalanced optimization: the performance in some environments is good while in other environments has inferior results. Consequently, disentangled preference representations via  $\mathbf{W}_z$  and  $\mathbf{W}_x$  might not be reliable across multiple environments. As such, we incorporate the variance penalty regularizer used in invariant learning [20, 27], which regulates the variance of the gradients under  $T$  environments. Specifically, we calculate the variance regularization for user  $u$  by  $\sum_{t=1}^T \|\nabla_{\theta} \mathcal{L}_t^u - \nabla_{\theta} \mathcal{L}^u\|^2$ , where  $\mathcal{L}_t^u$  is the optimization loss for the environment  $t$  in Eq. (9),  $\nabla_{\theta}$  denotes the gradients *w.r.t.* the learnable parameters  $\theta$ , and  $\nabla_{\theta} \mathcal{L}^u$  represents the average gradients over  $T$  environments.

Intuitively, the variance regularizer will restrict the gradient difference among  $T$  environments, and thus update the parameters  $\theta$  synchronously for multiple environments. This will alleviate the problem that the parameters are unfairly optimized to improve the performance of few environments [27]. To sum up, we have the final optimization loss for user  $u$  as follows:

$$\mathcal{L}^u + \lambda_2 \cdot (\|\mathbf{W}_z\|_0 + \|\mathbf{W}_x\|_0) + \lambda_3 \cdot \sum_{t=1}^T \|\nabla_{\theta} \mathcal{L}_t^u - \nabla_{\theta} \mathcal{L}^u\|^2, \quad (9)$$

where two hyper-parameters  $\lambda_2$  and  $\lambda_3$  control the strength of sparsity and variance regularization terms, respectively.

**2.2.4 Environment division.** For the temporal interaction sequence of user  $u$ , we split it into  $T$  pieces according to the equal interaction number in every environment. The choice of  $T$  is essential because it balances the learning of shifted and invariant preference. CDR only considers the cross-environment preference shifts and assumes the intra-environment preference is invariant by ignoring the temporal information of interactions. Therefore, a larger  $T$  will expose more sequential information to CDR. Nevertheless, the large  $T$  value will increase the sparsity of interactions in each environment, hurting the learning of the encoder and decoder networks. To alleviate the dilemma, we choose a relatively small  $T$  during training to ensure the interaction density of each environment. Once the parameters are well learned, we adopt a larger  $T$  to fully utilize the sequential information in the inference stage.

**2.2.5 Summary.** The detailed training procedure can be found in Algorithm 1. To train the encoder and decoder networks in CDR, we divide the training interactions into multiple environments and utilize them to minimize the loss function in Eq. (9) over all users. During the inference stage, we use the latest user preference  $\mathbf{z}_T$  to calculate  $f_{\theta_2}(\mathbf{z}_T)$  for the ranking of item candidates, and then recommend top-ranked items to each user.

To summarize, the encoder network infers unobserved  $E_t$  from users' interactions. Thereafter, the decoder network leverages the inferred  $E_t$  to iteratively update  $Z_t$  for better preference estimation. Besides, the decoder network conducts sparse structure learning to model the sparse influence from  $Z_t$  to  $X_t$  for better effect estimation of user preference. As compared to traditional VAE-based methods [25, 53, 66], CDR is more robust in OOD environments because it constructs the encoder and decoder networks by following the causal relations in Figure 2. Besides, thanks to modeling causal relations, CDR supports the intervention over the causal graph. As illustrated in Section 3.4.2, we can estimate the counterfactual user preference  $Z_t$  and the corresponding recommendations by intervening on  $E_t$ , *i.e.*, changing  $E_t = \mathbf{e}_t$  to  $do(E_t = \hat{\mathbf{e}}_t)$  [34].

**Algorithm 1** Training of CDR under Multiple Environments

---

**Input:**  $X_{1:T}$  of all  $U$  users;  $g_\phi(\cdot)$ ,  $f_{\theta_1}(\cdot)$ , and  $f_{\theta_2}(\cdot)$  with initialized  $\phi$ ,  $\theta_1$ , and  $\theta_2$ , respectively.

```

1: while not converged do
2:   Sample a batch of users  $\mathcal{U}$ 
3:   for all  $u \in \mathcal{U}$  do
4:     for all  $t \in \{1, 2, \dots, T\}$  do
5:       Sample  $\mathbf{e}_t$  by feeding  $\mathbf{x}_t$  into  $g_\phi(\mathbf{x}_t)$ ;
6:       Sample  $\mathbf{z}_t$  by feeding  $\mathbf{z}_{t-1}$  and  $\mathbf{e}_t$  into  $f_{\theta_1}(\mathbf{z}_{t-1}, \mathbf{e}_t)$ ;
7:       Calculate  $f_{\theta_2}(\mathbf{z}_t)$  via Eq. (7);
8:       Obtain the probability of drawing  $\mathbf{x}_t$  by Eq. (8);
9:       Calculate the gradients w.r.t. the loss in Eq. (9);
10:    end for
11:  end for
12:  Average the gradients over  $|\mathcal{U}|$  users and  $T$  environments;
13:  Update  $\phi$ ,  $\theta_1$ , and  $\theta_2$  via Adam;
14: end while

```

**Output:**  $g_\phi(\cdot)$ ,  $f_{\theta_1}(\cdot)$ , and  $f_{\theta_2}(\cdot)$ .

---

### 3 EXPERIMENTS

In this section, we conduct extensive experiments on three public datasets to answer the following research questions:

- **RQ1:** How does CDR perform under user preference shifts as compared to the baselines?
- **RQ2:** How can the different designs in CDR (e.g., the environment numbers, the sparse structure, inference strategies, multi-objective loss, and hyper-parameters) affect the performance?
- **RQ3:** How can we intuitively understand the effectiveness of CDR by case studies?

#### 3.1 Experimental Settings

- **Datasets.** We evaluate the baselines and the proposed CDR on three real-world datasets: 1) Yelp<sup>3</sup> is a public restaurant recommendation dataset, which contains rich interaction features such as ratings and timestamps; 2) Book is one of the Amazon product review datasets<sup>4</sup>, which covers extensive users' ratings over books; and 3) Electronics is also from the Amazon datasets, in which users interact with various electrical products.

The statistics of datasets are summarized in Table 1. To ensure the data quality [10], we only keep the users and items with at least 20 interactions on Yelp and Book. Besides, we only discard the users and items with less than 10 interactions on Electronics because the numbers of users and items are relatively small as shown in Table 1. Moreover, only the interactions with ratings  $\geq 4$  are considered as positive samples on all three datasets. We sort user interactions chronologically, and then split the interactions of each user by the ratio of 80%, 10%, 10% into training, validation, and test sets, respectively.

- **Baselines.** We compare CDR with the state-of-the-art collaborative filtering, disentangled and sequential models.

- **MF** [37] is one of the most influential collaborative filtering methods, which factorizes the sparse interaction matrix into the user and item embedding matrices.

<sup>3</sup><https://www.yelp.com/dataset>.

<sup>4</sup><https://jmcauley.ucsd.edu/data/amazon/>.

Table 1. Statistics of the three datasets.

Dataset	#User	#Item	#Interaction	Density
Yelp	11,622	9,095	487,000	0.004607
Book	21,923	23,773	1,125,676	0.002159
Electronics	9,279	6,065	158,979	0.002825

- **LightGCN** [10] is a powerful GCN-based recommender model, which discards the useless feature transformation and nonlinear activation in GCN, and highlights the most essential neighborhood aggregation for collaborative filtering.

- **MultiVAE** [23] is the most representative VAE-based recommender model, which captures the interaction generation process but ignores the temporal preference shifts.

- **MacridVAE** [31] proposes the disentangled user representations at the intention and preference levels. The disentanglement enhances the model robustness against preference shifts.

- **DIB** [28] utilizes information theory to disentangle biased and unbiased embeddings, and only considers unbiased embeddings for robust interaction prediction.

- **COR** [57] proposes a causal OOD framework to handle the observed user feature shifts.

- **DIEN** [82] focuses on a new structure to model the interest evolving process, leading to more expressive user representations.

- **MGS** [21] applies a session graph generated from the user interaction sequence to capture transition patterns of user preference. We do not use user and item features in COR, DIEN, and MGS for fair comparison with other methods.

- **DSSRec** [32] introduces the techniques of self-supervised learning and disentangled representations to sequential recommendation. However, it ignores the advantages of learning disentangled representations from multiple environments.

- **ACVAE** [67] is one of the state-of-the-art sequential recommender models, which incorporates contrastive learning and adversarial training into the VAE-based method.

- **CauseRec** [76] constructs counterfactual sequences by keeping the indispensable interactions and replacing the dispensable ones. These counterfactual sequences are then used as augmented samples for contrastive training.

We omit more sequential models such as GRU4Rec [13] and BERT4Rec [48] since ACVAE and CauseRec have shown better performance than them.

• **Evaluation.** We follow the all-ranking protocol [10] to evaluate the performance of all methods, where all non-interacted items are used for ranking and top-ranked items are returned as recommendations. Thereafter, we adopt Recall@K (R@K) and NDCG@K (N@K) as the evaluation metrics, where  $K = 10$  or  $20$  on three datasets.

• **Hyper-parameter settings.** Based on the default settings of baselines, we enlarge their hyper-parameter search scope and tune hyper-parameters as follows:

- MF & LightGCN: The learning rate is searched in  $\{0.001, 0.01, 0.1\}$ . We search the best embedding size from  $\{32, 64, 128\}$ . For LightGCN, we tune the weight decay in  $\{1e^{-5}, 1e^{-4}, 1e^{-3}\}$  and the number of GCN layers in  $\{3, 4, 5\}$ .

- MultiVAE & MacridVAE: We follow the default settings and additionally tune the learning rate, the hidden size, the regularization coefficient  $\beta$  in  $\{1e^{-4}, 1e^{-3}, 1e^{-2}\}$ ,  $\{[800], [600, 200], [800, 500]\}$ , and  $\{0.3, 0.5, 0.7, 0.9\}$ , respectively. As to special hyper-parameters in MacridVAE, we choose the number of macro factors from  $\{2, 4, 10, 20\}$ , the number of micro factors from  $\{200, 300, 400, 500\}$ , and the coefficient  $\tau$  from  $\{0.05, 0.1, 0.2\}$ .

- DIB & COR: For DIB, the embedding size is set in  $\{32, 64, 128\}$ . We adjust  $\alpha$  in  $\{0.001, 0.1, 0\}$ ,  $\beta$  in  $\{0.0001, 0.001, 0.01, 0.1\}$  and  $\gamma$  in  $\{0, 0.1, 0.2\}$  to make a balance between the biased and unbiased vector. For COR, we

Table 2. The performance comparison between the baselines and CDR on the three datasets. The best results are highlighted in bold and the second-best ones are underlined. %improve. indicates the relative improvements of CDR than the second-best results. \* implies the improvements over the best baseline are statistically significant ( $p$ -value  $< 0.05$ ) under one-sample t-tests.

Methods	Yelp				Book				Electronics			
	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20	R@10	R@20	N@10	N@20
MF	0.0385	0.0659	0.0269	0.0365	0.0206	0.0355	0.0142	0.0193	0.0333	0.0551	0.0186	0.0243
LightGCN	0.0402	0.0695	0.0288	0.0390	0.0252	0.0434	0.0170	0.0233	0.0365	0.0560	0.0194	0.0246
MultiVAE	0.0427	0.0728	0.0303	0.0409	0.0280	0.0475	0.0197	0.0264	0.0419	0.0658	0.0224	0.0286
MacridVAE	<u>0.0442</u>	<u>0.0770</u>	0.0319	0.0434	0.0409	0.0667	0.0288	0.0378	0.0424	0.0654	<u>0.0246</u>	0.0306
DIB	0.0375	0.0654	0.0264	0.0362	0.0211	0.0362	0.0144	0.0196	0.0319	0.0530	<u>0.0176</u>	0.0231
COR	0.0411	0.0690	0.0293	0.0392	0.0400	0.0681	0.0286	0.0385	0.0428	0.0625	<u>0.0237</u>	0.0289
DIEN	0.0275	0.0449	0.0202	0.0263	0.0279	0.0382	0.0250	0.0284	0.0347	0.0512	<u>0.0196</u>	0.0240
MGS	0.0423	0.0696	0.0313	0.0409	0.0515	0.0738	0.0458	0.0532	0.0415	0.0625	<u>0.0230</u>	0.0285
DSSRec	0.0413	0.0697	0.0299	0.0400	0.0539	0.0790	0.0448	0.0534	<u>0.0503</u>	<u>0.0780</u>	0.0269	0.0343
ACVAE	0.0439	0.0750	<u>0.0322</u>	0.0432	<u>0.0563</u>	<u>0.0860</u>	<u>0.0477</u>	<u>0.0576</u>	<u>0.0510</u>	<u>0.0766</u>	<u>0.0290</u>	<u>0.0359</u>
CauseRec	0.0433	0.0762	0.0300	0.0417	0.0484	0.0753	0.0391	0.0482	0.0445	0.0744	0.0230	0.0309
CDR	<b>0.0528*</b>	<b>0.0880*</b>	<b>0.0392*</b>	<b>0.0518*</b>	<b>0.0721*</b>	<b>0.1042*</b>	<b>0.0598*</b>	<b>0.0708*</b>	<b>0.0647*</b>	<b>0.0933*</b>	<b>0.0373*</b>	<b>0.0449*</b>
% Improve.	19.46%	14.29%	21.74%	19.35%	28.06%	21.16%	25.37%	22.92%	26.86%	19.62%	28.62%	25.07%

search the hidden size of encoder  $q(\cdot)$  and the KL coefficient  $\beta$  in  $\{[300], [800], [800, 600]\}$ , and  $\{0.3, 0.5, 0.7, 0.9\}$ , respectively. The sizes of  $Z_1$  and  $Z_2$  are chosen in  $\{100, 200, 300\}$ .

- DIEN & MGS: For DIEN, the embedding size is set in  $\{32, 64, 128, 256, 512\}$ . We search the dropout ratio in  $\{0.1, 0.3, 0.5, 0.6\}$  and the weight of auxiliary loss in  $\{0, 0.4, 0.6, 0.8, 1.0\}$ . For MGS, we choose the embedding size from  $\{50, 100, 200\}$ , the number of GNN layers from  $\{1, 3, 5, 6\}$ , and the sequence length from  $\{10, 30, 50\}$ .

- DSSRec & ACVAE & CauseRec: We search the training sequence length and the embedding size in  $\{50, 100, 200\}$  and  $\{32, 64, 128\}$ , respectively. For DSSRec, the number of latent categories  $K$  is chosen from  $\{1, 2, 4, 8\}$  and the weight decay is set by  $\{0, 0.01, 0.05\}$ . For ACVAE, the weight of contrastive loss term  $\beta$  is searched in  $\{0.1, 0.3, 0.5, 0.7\}$ . For CauseRec, the number of concepts and weight decay are tuned in  $\{20, 30, 40\}$  and  $\{1e^{-5}, 1e^{-4}, 1e^{-3}\}$ , respectively.

As to CDR, we implement it by Pytorch and utilize Adam for optimization. For fair comparison, we choose the hyper-parameters by following the settings of the baselines. We set the batch size as 500 and the learning rate as  $1e^{-4}$ . The dropout ratio is chosen from  $\{0.4, 0.5, 0.6\}$ . The hidden size of  $g_\phi(\cdot)$  is searched in  $\{[800], [600, 200], [800, 500]\}$ . The sizes of  $e_t$  and  $z_t$  are tuned in  $\{200, 300, 400, 500\}$ . Both  $f_{\theta_1}(\cdot)$  and  $f_\gamma(\cdot)$  are set as a fully-connected layer to save parameters.  $T$ ,  $\lambda_1$ ,  $\lambda_2$ ,  $\lambda_3$  and  $C$  are tuned in  $\{1, 2, \dots, 6\}$ ,  $\{0.1, 0.2, \dots, 0.9\}$ ,  $\{0.1, 0.2, \dots, 1\}$ ,  $\{1e^{-5}, 1e^{-4}, 1e^{-3}, 1e^{-1}\}$ , and  $\{1, 2, 3, 4, 10, 20\}$ , respectively. Moreover, early stopping is performed for model selection, *i.e.*, stop training if recall@10 on the validation set does not increase for 10 successive epochs. More details can be found in the released code.

### 3.2 Overall Performance (RQ1)

We present the results of the baselines and CDR on the three datasets in Table 2. From the table, we have the following observations:

- MacridVAE consistently outperforms MF, LightGCN, and MultiVAE on the three datasets. We attribute the superior performance to the disentangled user representations of MacridVAE. The preference shifts only affect

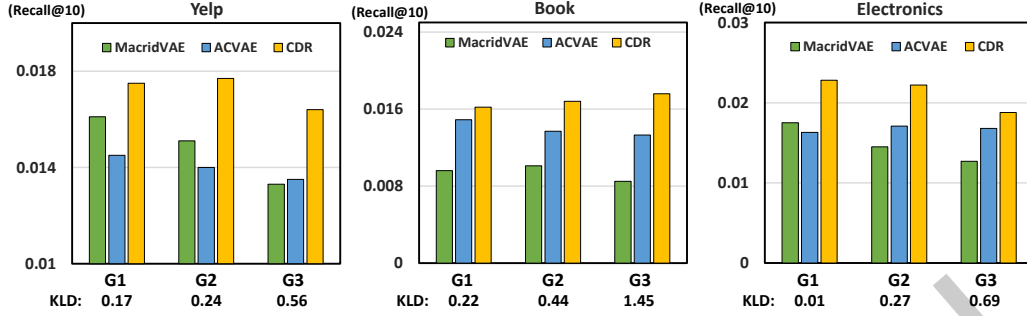


Fig. 5. Performance comparison on three user groups with different strengths of shifts, where the shifts increase from G1, G2, to G3.

partial user representations while most disentangled user representations of MacridVAE are robust to the shifts. Besides, the sequential models (*i.e.*, DSSRec, ACVAE, and CauseRec) and the session-based model (*i.e.*, MGS) usually perform better than MF, LightGCN, and MultiVAE, which verifies the effectiveness of considering temporal information in capturing preference shifts.

- The performance of MacridVAE is better than that of sequential models (DSSRec, ACVAE, and CauseRec) on Yelp while the sequential models surpass MacridVAE on Book and Electronics. Meanwhile, DIEN outperforms MF on Book and Electronics while yields inferior performance on Yelp. This is because the effect of preference shifts is quite different on the three datasets. Temporal preference shifts are stronger on Book and Electronics, and thus sequential modeling is more effective to capture the shifts. In contrast, user preference over food is relatively stable on Yelp, where MacridVAE is superior to model the invariant preference.
- In sequential models, ACVAE usually achieves higher performance than DSSRec and CauseRec. This is probably because ACVAE introduces adversarial training and contrastive learning into sequential VAE, which also encourages the independence of latent factors in user representations. Such independence might have a similar effect as disentangled representations of CDR, *i.e.*, the sparse structure. The main difference is that the disentangled representations of CDR are learned from multiple environments, which are more robust under preference shifts. Besides, the inferior performance of DSSRec and CauseRec might be attributed to the improper intention clustering [32] and inaccurate identification of dispensable concepts [76], respectively.
- COR is usually comparable with MacridVAE on Book and Electronics while performs worse than sequential models such as ACVAE. This is reasonable since COR eliminates the out-of-date information and reuses the stable preference, leading to robust user representations against user preference shifts. However, COR ignores the temporal feature shifts, resulting in worse performance than sequential models. Besides, DIB shows relatively worse results, which is possible because that biased embeddings might be still useful and totally discarding them loses critical user preference.
- CDR significantly yields the best performance on the three datasets. Specifically, the performance improvements of CDR over the best baseline *w.r.t.* Recall@10 are 19.46%, 28.06%, and 26.86% on Yelp, Book, and Electronics, respectively. This justifies the superiority of handling user preference shifts via the CDR framework. CDR does not only capture the temporal preference trend between environments for better preference estimation, but also learns a robust structure from user preference to interactions, leading to better interaction prediction.

To evaluate the performance of CDR under different strengths of shifts, we split users into groups according to the KL divergence between training and testing environments *w.r.t.* their interacted item categories. As shown in Figure 5, the preference shifts increase from G1, G2, to G3. From the figure, we can find that 1) CDR consistently

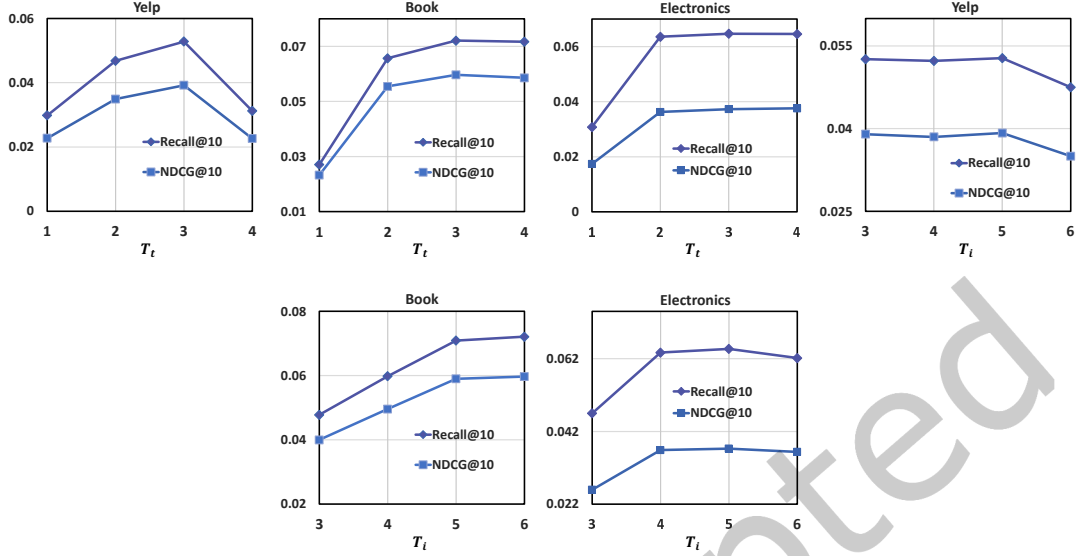


Fig. 6. Effect of environment numbers in the training ( $T_t$ ) and inference ( $T_i$ ) stages. We obtain the results of different  $T_i$  by using the best  $T_t = 3$  for training and vary  $T_i$  for inference.

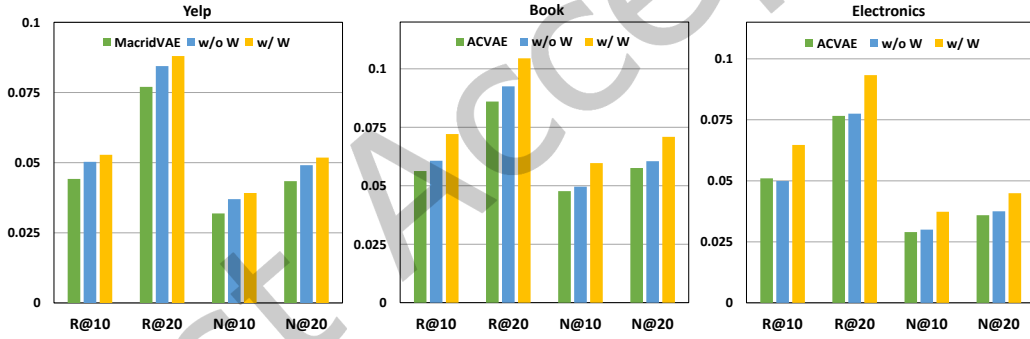


Fig. 7. Ablation study of the sparse structure (*i.e.*,  $W_z$  and  $W_x$ ). We only show the best baseline on each dataset to save space.

achieves better performance across three groups; and 2) the performance of the best baselines, MacridVAE and ACVAE, usually decreases in the G3 group with large shifts while CDR still shows large improvements over the baselines, validating its stronger OOD generalization ability.

### 3.3 In-depth Analysis (RQ2)

In addition to overall performance comparison, we conduct the in-depth analysis to study the effectiveness of different components in CDR, including multiple environments, the sparse structure for the disentanglement, conditional relations, inference strategies, the multi-objective loss, and hyper-parameter settings. Lastly, we provide some cases to show the effectiveness of CDR at a fine-grained level.

**3.3.1 Effect of Multiple Environments.** As illustrated in Section 2.2.4, the choice of the environment number  $T$  is essential. As such, we study the effect of multiple environments by varying the environment numbers in the

Table 3. Performance comparison of CDR with and without conditional relations.

	Yelp		Book		Electronics	
	R@10	N@10	R@10	N@10	R@10	N@10
ACVAE	0.0439	0.0322	0.0563	0.0477	0.0510	0.0290
CDR with $E_{t-1} \rightarrow E_t$	0.0450	0.0335	0.0584	0.0466	0.0549	0.0306
CDR with $X_{t-1} \rightarrow Z_t$	0.0458	0.0340	0.0572	0.0491	0.0553	0.0314
Vanilla CDR	0.0528	0.0392	0.0721	0.0598	0.0647	0.0373

training and inference periods. We report the results with different environment numbers during training ( $T_t$ ) and inference ( $T_i$ ) in Figure 6. From the figure, we have the following findings.

- During training, the performance rises at first, and then drops with the increase of  $T_t$ . The rise validates the effectiveness of learning user representations from multiple environments instead of one environment. Besides, the performance drop also verifies the arguments in Section 2.2.4: more environments will make the interactions in each environment sparse, hindering the learning of invariant user preference within an environment and the disentangled representations (*i.e.*, the sparse structure  $\mathbf{W}_z$  and  $\mathbf{W}_x$ ).
- In the inference stage, a larger  $T_i$  than  $T_t$  is able to further improve the performance of CDR, especially on Book and Electronics. This is due to the better utilization of temporal information. During inference, the sparsity issue in each environment will not affect the optimization of CDR. As such, we can fully utilize the temporal information of interactions and consider more fine-grained temporal preference shifts by using larger  $T_i$  for the inference.
- The effectiveness of increasing  $T_i$  is more significant on Book and Electronics than that of Yelp. The underlying reason is that Book and Electronics have stronger temporal shifts as discussed in Section 3.2 and more environments help to capture more fine-grained shifts between environments. In contrast, user preference is relatively stable on Yelp, and thus CDR leverages fewer environments to better capture the invariant preference within each environment. This shows that CDR can flexibly balance the learning of invariant and shifted preference on different datasets by adjusting the environment number.

**3.3.2 Effect of Sparse Structure.** To validate the effectiveness of the sparse structure from user preference to interactions, we perform the ablation studies over the two matrices  $\mathbf{W}_z$  and  $\mathbf{W}_x$ . The results with (*i.e.*, w/  $\mathbf{W}$ ) and without (*i.e.*, w/o  $\mathbf{W}$ ) the two matrices are provided in Figure 7. The ablation of  $\mathbf{W}_z$  and  $\mathbf{W}_x$  denotes that CDR only uses an MLP model  $f_y(\cdot)$  to obtain  $f_{\theta_2}(z_t)$  in Eq. (7). From Figure 7, we can observe that:

- The performance declines if the two matrices are removed, showing the effectiveness of the sparse structure in modeling the effect of user preference shifts.
- CDR without  $\mathbf{W}_z$  and  $\mathbf{W}_x$  still outperforms the best baselines, *i.e.*, MacridVAE and ACVAE. Such improvements are attributed to the division of environments: without the disentanglement via the sparse structure, CDR still captures both the preference shifts between environments and the invariant preference within each environment by following the robust causal relations [1, 29].

**3.3.3 Effect of Conditional Relations.** We conduct experiments to compare the CDR performance with and without considering the conditional relations of  $E_{t-1} \rightarrow E_t$  and  $X_{t-1} \rightarrow Z_t$ . Considering them will change the encoder  $q(\mathbf{e}_t|\mathbf{x}_t)$  and the decoder module  $p(\mathbf{z}_t|\mathbf{e}_t, \mathbf{z}_{t-1})$  into  $q(\mathbf{e}_t|\mathbf{x}_t, \mathbf{e}_{t-1})$  and  $p(\mathbf{z}_t|\mathbf{e}_t, \mathbf{z}_{t-1}, \mathbf{x}_{t-1})$ , respectively. Such changes introduce more parameters due to the larger input dimension. The experimental results are presented in Table 3, from which we observe that 1) CDR with  $E_{t-1} \rightarrow E_t$  or  $X_{t-1} \rightarrow Z_t$  has inferior performance than the vanilla CDR. The possible reasons are that these conditional relations are not strong over a large proportion of users, and meanwhile the CDR with more parameters might overfit the training data, hurting the generalization

Table 4. Performance comparison of three inference strategies.

	Yelp		Book		Electronics	
	R@10	N@10	R@10	N@10	R@10	N@10
1) $z_T$	0.0528	0.0392	0.0721	0.0597	0.0647	0.0373
2) avg $x_{1:T}$	0.0429	0.0312	0.0288	0.0206	0.0476	0.0258
3) $z_{T+1}$	0.0470	0.0340	0.0550	0.0449	0.0613	0.0336

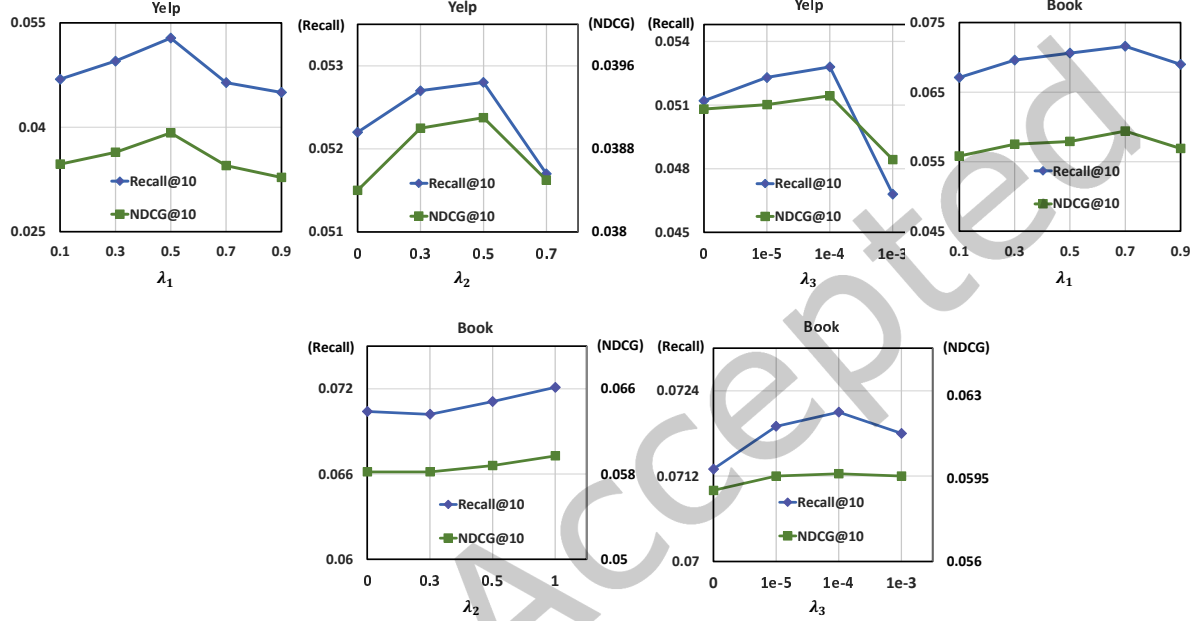


Fig. 8. Effect of the coefficients in the multi-objective loss (i.e.,  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ ). Specifically,  $\lambda_1$  is the coefficient of KL divergence;  $\lambda_2$  controls the sparsity regularization; and  $\lambda_3$  adjusts the variance penalty term.

ability in a new environment. Besides, 2) CDR with  $E_{t-1} \rightarrow E_t$  or  $X_{t-1} \rightarrow Z_t$  still surpasses the best baseline ACVAE, validating the effectiveness of modeling preference shifts and sparse influence by this CDR framework.

**3.3.4 Effect of Inference Strategies.** In Table 4, we report the results of the three inference strategies detailed in Section 2.1.2. From Table 4, we can find that 1) the first strategy outperforms the second and third strategies, and 2) the second one shows the worst results. These findings are reasonable because 1) the second strategy ignores the temporal distribution shifts and uniformly averages the predictions, and 2) the third strategy is better since it partly considers the shifts by  $z_T$  while  $e_{T+1}$  is still obtained by average, inevitably losing some temporal patterns. In future work, it is promising to explore more strategies to better capture temporal patterns for the inference.

**3.3.5 Effect of Multi-objective Loss.** To analyze the influence of KL divergence, the sparsity, and variance regularizers in the multi-objective loss, we conduct experiments to compare the performance by changing their coefficients  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$ . The results *w.r.t.* the three coefficients on Yelp and Book are presented in Figure 8. The results on Electronics and Book are similar so that we omit the one on Electronics. From Figure 8, we have the following observations:



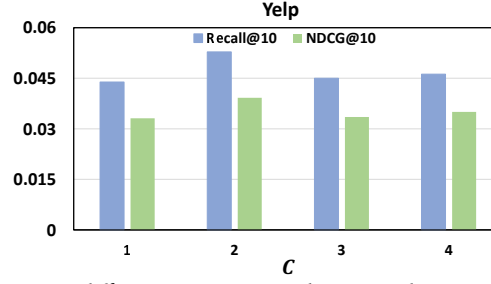


Fig. 9. Effect w.r.t. different category numbers  $C$  in the structure learning.

- Increasing  $\lambda_1$  of KL divergence is useful to improve the performance. Actually, as indicated by [14], a large  $\lambda_1$  for KL divergence regulates the independence of latent factors to facilitate disentangled representations; while a small  $\lambda_1$  helps the model to fit the user interactions better [25]. From Figure 8, we find that  $\lambda_1 \in [0.5, 0.7]$  usually leads to a good balance and achieves superior performance, which is consistent with the findings in [25].
- The sparsity regularizer is essential to improve the performance because Recall and NDCG drop significantly when  $\lambda_2$  is changed from 0.5 to 0. Besides,  $\lambda_2$  cannot be too large, which will limit the representation capability of  $\mathbf{W}_z$  and  $\mathbf{W}_x$ .
- The decreased performance from  $\lambda_3 = 1e^{-4}$  to  $\lambda_3 = 0$  justifies the usefulness of the variance regularizer, which balances the gradient optimization across multiple environments. Moreover, it should be noted that  $\lambda_3$  is expected to be tuned in a small magnitude  $[0, 1e^{-3}]$  because the stronger gradient regularization will inevitably disturb the normal optimization of parameters.

**3.3.6 Effect of Category Number.** To verify the effect of category numbers in sparse structure learning, we compare the performances with  $C$  changing from 1 to 4. Note that  $C = 1$  is equivalent to removing the sparse structure, i.e., using pure MLP to predict the interaction probability based on  $\mathbf{z}_t$ . We report the performance of CDR in Figure 9. By comparing the results w.r.t. Recall@10 and NDCG@10, we can find that:

- The inferior performance of  $C = 1$  justifies that disentangling user representations into categorical-level preference alleviates the negative effect of preference shifts. This is rational since it aligns with the real-world scenarios: items are classified into various categories and users have different preference over such categories. Once users have preference shifts, only partial factors in the user representations change and subsequently affect partial interactions, leading to better OOD generalization.
- The fluctuated performance from  $C = 2$  to  $C = 4$  indicates that the performance increase is not proportional to the category number and  $C = 2$  shows better results. However, we usually have more categories in the real-world scenarios, for example, a variety of books. This implies that 1) it is non-trivial to recover the category-level preference from pure interaction data; and 2) incorporating item category into recommender models might help the disentanglement, which is left to future exploration.

**3.3.7 Effect of Hidden Size.** We conduct experiments to investigate the effect of the hidden sizes of user representations and the latent dimension of VAE networks. In particular, we present the results w.r.t. varying hidden sizes of  $\mathbf{e}_t$  ( $K$ ),  $\mathbf{z}_t$  ( $H$ ), and the decoder network  $g_\phi(\cdot)$ . From the results reported in Figure 10, we have the following findings:

- When the hidden size of  $g_\phi(\cdot)$  changes from [200] and [400] to [800], we can observe that a wider  $g_\phi(\cdot)$  yields superior performance. Besides, by comparing the hidden sizes of  $\mathbf{e}_t$  and  $\mathbf{z}_t$  in the range  $\{200, 300, 400\}$ , we find a larger size significantly improves Recall@10 and NDCG@10. Such improvements are attributed to enriching the representation abilities of the encoder network and user representations.

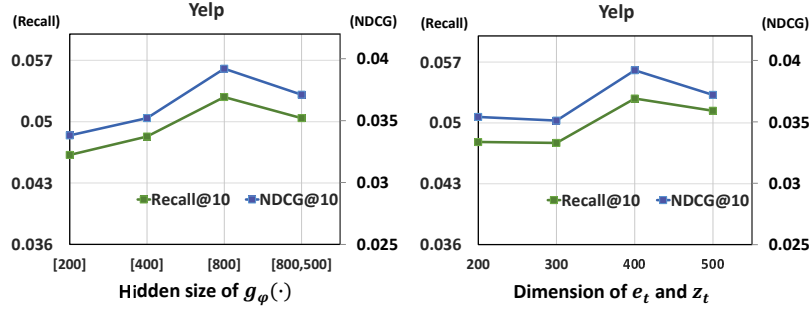
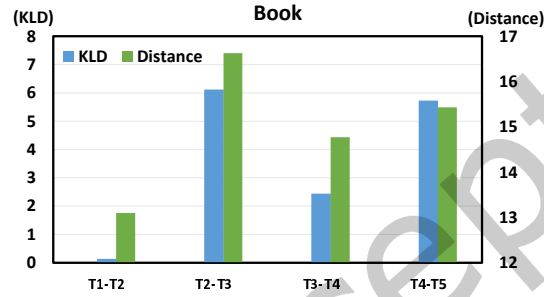
Fig. 10. Effect of the hidden sizes of  $g_\phi(\cdot)$ ,  $e_t$ , and  $z_t$ .

Fig. 11. Visualization of the KL divergence and Euclidean distance for the sampled user in Book. Note that KL divergence reflects the preference shifts in the interactions while Euclidean distance describes the similarity between the learned user representations.

- Nevertheless, the model will suffer from over-fitting issue if we blindly increase the number of parameters. For example, the performance drops as we add the layers of  $g_\phi(\cdot)$  from [800] to [800, 500]. Therefore, controlling the parameter number for better OOD generalization is a wise choice. We should carefully tune the hidden sizes to balance the trade-off between the representation ability and over-fitting issue.

We do not show the results on the decoder network because it is implemented by two full-connected layers (*i.e.*,  $f_{\theta_1}(\cdot)$  and  $f_{\lambda}(\cdot)$ ) whose dimension is decided by  $K$ ,  $H$ , and the item number  $I$  in the dataset. Similar to the encoder network  $g_\phi(\cdot)$ , we have validated that increasing the layer number of  $f_{\theta_1}(\cdot)$  and  $f_{\lambda}(\cdot)$  fails to improve the OOD generalization performance.

### 3.4 Case Study (RQ3)

**3.4.1 Alignment between User Representation and Shifted Preference.** To intuitively understand how CDR captures the preference shifts, we conduct the case study from the user level and population level, respectively. Specifically, for each user, we extract user preference representation  $z_t$  and the interaction distribution over item categories at each environment. Thereafter, we study whether the shifts of user presentations align with the category-level interaction distribution, which can reflect if the user representations capture preference shifts well.

- **User-level analysis.** We randomly select a user from the Book dataset who has significant shifts over the category-level interaction distribution. For the sampled user, we calculate the KL divergence between the category-level interaction distributions in every two continuous environments. Besides, we estimate the user representation similarity between continuous environments by Euclidean distance. The changes of KL divergence and Euclidean

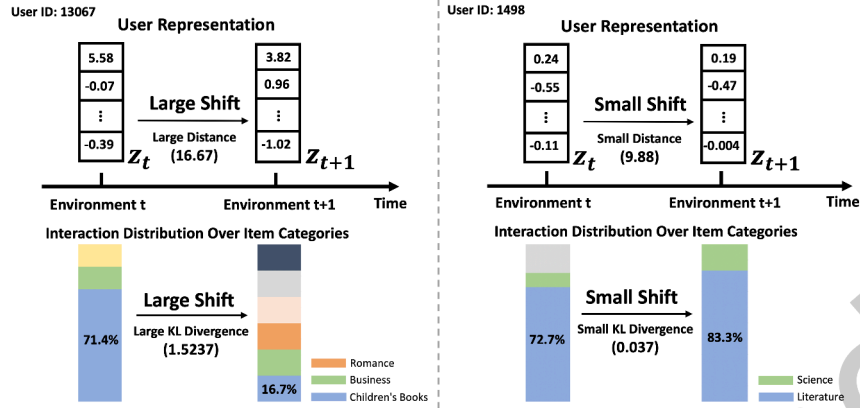


Fig. 12. Visualization of the interaction distributions and the learned representations for two users in Book. The left figure depicts the user's dramatic shift over category-level preference between two environments. By contrast, the user in the right figure shows relatively stable preference for book genres. The changes of user representations have a good alignment with the user preference shifts.

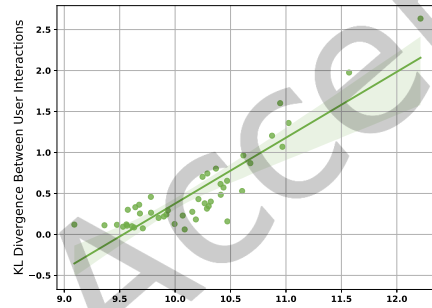


Fig. 13. Visualization of KL divergence of interactions and Euclidean distance of user representations at the population level, which shows a strong positive correlation. This demonstrates that the user representations learned by CDR well capture the interaction shifts.

distance in five environments are presented in Figure 11. Furthermore, we also visualize the specific interaction distribution of another two randomly selected users at a more fine-grained level, whose user representations and interaction distributions are provided in Figure 12. From the two figures, we have the following conclusions.

- The Euclidean distance between user representations has a consistent pattern with the KL divergence between the category-level interaction distributions. For example, when the KL divergence is large in Figure 11 (i.e., T1-T2 and T3-T4), the user representations show the large distance correspondingly. This indicates that user representations are less similar if the preference in the interactions is significantly shifted. In other words, such user representations are capable of capturing the preference shifts.
- As to the specific examples in Figure 12, we can find that the KL divergence well describes the preference shifts over categories: the first user's preference for Children's Books radically drops from 71.4% to 16.7% and we have a high KL divergence; by contrast, the interests of the second user are stable, and thus this user has a smaller KL divergence. More importantly, we observe that the user representations have the same distance shifts as the interaction distributions, which is consistent with the findings in Figure 11.

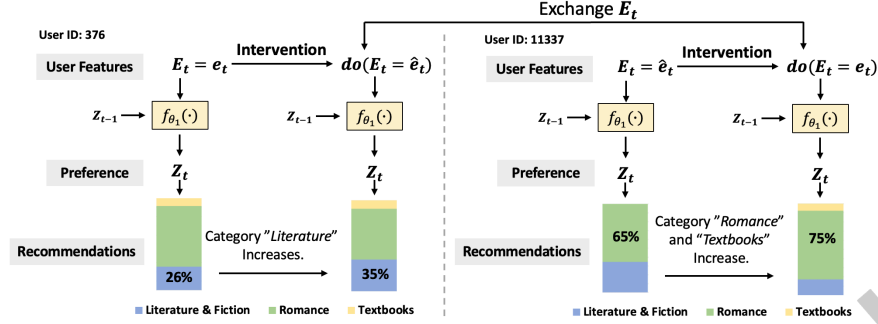


Fig. 14. Visualization of the recommendation changes when exchanging  $E_t$  of two users in Book. Accordingly, the recommendations become more similar to each other after the intervention.

• **Population-level analysis.** Although the user representations learned by CDR well capture the preference shifts for these users, how does CDR perform over the whole dataset? To answer this question, we do a population-level analysis. Specifically, we rank the users in Book according to the average KL divergence across multiple environments, and then divide all users into 50 groups based on the ranking. Next, we calculate the average KL divergence and the average Euclidean distance between user representations within each group. Visualization of the correlation between the KL divergence and Euclidean distance is shown in Figure 13, which validates that the Euclidean distance between user representations is proportional to the KL divergence between user interactions. This demonstrates that our proposed CDR learns the user representations well to capture the preference shifts in the whole dataset.

**3.4.2 Case Study on  $do(E_t = \hat{e}_t)$ .** We do another case study to inspect whether the do-operation over  $E_t$  generates reasonable changes in the recommendations. Specifically, we select two users in the Book dataset, exchange their user features  $E_t$ , and compare the changes of recommendations. Figure 14 shows that their recommendations become similar to each other after the intervention. More “Literature & Fiction” books are recommended to User 376 due to the high preference of User 11337 while some textbooks are exposed to User 11337 because User 376 likes them. These observations demonstrate that intervening  $E_T$  can affect the predictions of  $X_T$  and cause reasonable changes to recommendations in the CDR framework.

## 4 RELATED WORK

In this work, we propose a causal disentangled framework to handle the preference shifts in recommendation, which is closely related to causal recommendation, disentangled recommendation, sequential recommendation, and preference shifts in recommendation.

### 4.1 Causal Recommendation

In the past decade, data-driven recommender systems have been widely employed to alleviate the issue of information explosion on the Web [12, 57, 63]. Even though great success has been achieved, such data-driven approaches suffer from the issues of bias [79, 65], unfairness [6], and filter bubbles [8]. Recently, the emerging causal approaches shed light on them [2, 15, 46, 78, 87, 69]. Specifically, two strands of frameworks receive the most attention: the potential outcome framework [39] and structural causal models [34]. The potential outcome framework leverages inverse propensity scoring [42] or doubly robust [59] to address the problem of biases in explicit and implicit feedback [77], such as position bias [17] and exposure bias [42]. Regarding the structural causal models, intervention [55, 79] and counterfactual inference [61, 76, 87] are used to estimate the causal effects [35]

based on the causal relationships. And then the causal effects are more reliable to address recommendation issues, for example, debiasing, unfairness, OOD recommendation, and explanation [12, 24, 51, 56, 57].

Although causal reasoning has been widely applied to recommendation [7, 30, 68, 85] and some studies explore the OOD recommendation [12, 57], current methods usually utilize user and item features for generalization and ignore the temporal preference shifts across environments. To fill the gap, this work studies the under-explored temporal preference shifts without using extra user or item features. We discover the generation process of preference shifts under multiple environments and develop a causal framework to alleviate the detrimental effect of preference shifts.

## 4.2 Disentangled Recommendation

Disentangled recommendation learns the independent representations for the hidden factors (*e.g.*, user intention and preference) behind the complex user behaviors, which can bring various merits such as offering explanation or improving the model robustness in the drifted distributions [32, 35]. In order to capture users' diverse preference on items, previous work often disentangles user preference by encouraging the independence of user representations [32, 58, 60]. For instance, MacridVAE [31] identifies the high-level intention representations for macro disentanglement, and forces each individual dimension in the intention representation to be independent for micro disentanglement. Besides, Wang *et al.* proposed to disentangle the user representations in GCN-based recommender models to model the finer granularity of user intention. Lastly, MTIN [16] designs a time-aware mask network to distill the interaction sequence and adopts an interest mask network to aggregate fine-grained user preference representations.

However, previous studies typically learn disentangled representations from the IID data, and thus ignore the significance of capturing robustness across multiple environments [29], decreasing the generalization ability under user preference shifts. This work extends the disentangled approaches by considering both the multiple OOD environments and temporal preference shifts across environments. Additionally, we reformulate the disentangled representations in recommendation by two matrices and learn the matrices via the sparsity and variance regularization.

## 4.3 Sequential Recommendation

Collaborative filtering-based methods are widely employed in recommender systems [25, 43, 22, 82], where the user-item matching score is obtained based on user/item representation learned from user historical interactions. However, users' interactions are not independently generated because sequential patterns usually exist within users' consecutive behaviors. Therefore, sequential recommendation, which aims to recommend the next item to a target user, emerges and becomes popular in recent years [33, 36, 41, 84]. Early work utilizes Markov Chain to capture the lower-order dependencies [9, 38]. Later on, deep sequential models (*e.g.*, RNN [13, 83], CNN [50], Transformer [18, 48], GNN [3, 64] and others [40]) are employed to capture the higher-order dependencies. In addition to general sequential methods, CauseRec [76] conditionally constructs the counterfactual interaction sequences, and then performs contrastive representation learning by using both counterfactual and observational data. DSSRec [32] disentangles the intentions behind the user interaction sequence, and constructs seq2seq training samples by using only pairs of sub-sequences with the same intention, leading to better sequential modeling. Recently, ACVAE [67] has introduced adversarial learning to the variational Bayes framework for sequential recommendation and utilized contrastive learning to learn better user representations.

Different from sequential recommendation, CDR focuses on the generalizable recommendation to handle preference shifts, which aims to predict the user preference in the new OOD environment instead of next-item recommendation. Technically, CDR learns the invariant user preference in a short period while capturing temporal

preference shifts between multiple environments. Furthermore, CDR formulates a sparse structure from the preference representation to interaction prediction for more robust disentanglement.

#### 4.4 Preference Shifts in Recommendation

User preference may shift over time for many reasons, including the changes of user features (*e.g.*, income and pregnancy) and environment factors (*e.g.*, seasonal variation). For example, in the scenario of food recommendation, a user might become liking expensive but healthy food if the user's income increases. Since such preference shifts are frequent in the real-world scenarios, the recommender models should update the user representations adaptively over time. Ignoring the shifting nature of preference to learn user representation will lead to inappropriate recommendations. In addition to disentangled recommendation and sequential recommendation, there exists some work with the potential of addressing this issue. In particular, Aspect-MF [75] analyses the dynamicity of temporal preference aspects using a component-based approach, and identifies the aspects that are easy to drift. ST-LDA [72] learns region-dependent personal interests and crowd preference to adapt to preference shifts. Lastly, MTUPD [62] utilizes a forgetting curve function to calculate the correlations of user preference in different time periods.

- **Long- and short-term interest modeling.** Another possibly related direction considers the modeling of both long- and short-term interests, where short-term interests are inferred from recent interactions to capture the preference shifts, and long-term interests represent the stable preference over time. Many studies have shed light on the modeling of long- and short-term user interests, for instance, PLSPL [47] utilizes attention mechanism to characterize the long-term preference while integrating the location and category information to capture the short-term preference. LSTPM [49] develops a context-aware nonlocal network structure to explore the temporal and spatial correlations in users' trajectories for the long-term preference, and adopts a geo-dilated RNN to fully exploit the geographical relations for the short-term preference. Lastly, KERL [54] fuses knowledge graph into a reinforcement learning framework to capture long-term preference and predict short-term interests. Although the short-term preference might infer the preference shifts, such work has not discovered the causal reasons for invariant and shifted preference. By contrast, we model the invariant and shifted preference simultaneously by leveraging their underlying causal relations and conduct the disentangled preference learning from multiple OOD environments.
- **Domain adaptation.** Domain adaptation has been widely applied to solve the problem of distribution shifts, which improves the adaptation ability by using less data. In recommendation, its main application scenarios include cross-domain recommendation [80, 81, 86] and cold-start recommendation [74]. Technically, model adaptation can be implemented by parameter patch [45, 74], feature transformation [26], and meta learning [73]. These techniques have been well studied to estimate the preference of the users in a new domain or the cold-start users within the same domain. However, the preference shift in OOD recommendation is totally different because it is related to the same users with dynamic interests over time. This requires us to consider the connection between the interactions in different environments, *i.e.*, the cross-environment preference learning.

## 5 CONCLUSION AND FUTURE WORK

In this work, we formulated the preference shifts from a causal view and inspected the underlying causal relations from the perspective of multiple environments. Based on the causal relations, we proposed the CDR framework, which captures the preference shifts across environments via a temporal VAE and learns the sparse structure from user preference to interactions. In particular, CDR leverages two learnable matrices to disentangle user representations and formulate the sparse structure. We optimized CDR by a multi-objective loss with variance and sparsity regularization. During the training and inference stages, we could flexibly adjust the number of environments to balance the learning of shifted and invariant preference on different datasets. Extensive

experiments validate the effectiveness of CDR in capturing preference shifts and achieving superior generalization performance than the baselines. Furthermore, the in-depth analysis demonstrates the rationality of using multiple environments, the sparse structure, and the multi-objective loss.

This work attempts to learn user preference from multiple environments for handling user preference shifts. In this light, there are many promising directions in future work. In particular, 1) as discussed in Section 2.2.4, we equally split user interactions into  $T$  environments to simplify the data pre-processing. Nevertheless, it can be improved by developing more effective but complex methods to divide environments, e.g., clustering interactions according to the time interval. 2) It is meaningful to discover more fine-grained causal relations in Figure 2, such as the mutual impact between the category-level user preference. Besides, we might utilize observed user features (e.g., age) to study more fine-grained relations between  $E_t$  and  $Z_t$ . And 3) it is non-trivial to incorporate item features for the disentangled representation learning, which may help CDR to distinguish the category-level user preference.

## REFERENCES

- [1] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Nan Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher J. Pal. 2020. A Meta-Transfer Objective for Learning to Disentangle Causal Mechanisms. In *ICLR*.
- [2] Stephen Bonner and Flavian Vasile. 2018. Causal Embeddings for Recommendation. In *RecSys*. ACM, 104–112.
- [3] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential Recommendation with Graph Neural Networks. In *SIGIR*. ACM, 378–387.
- [4] Ming-Hui Chen, Qi-Man Shao, and Joseph G Ibrahim. 2012. *Monte Carlo methods in Bayesian computation*. Springer Science & Business Media.
- [5] Chung, Junyoung and Kastner, Kyle and Dinh, Laurent and Goel, Kratarth and Courville, Aaron C and Bengio, Yoshua. 2015. A recurrent latent variable model for sequential data. In *NeurIPS*, Vol. 28. Curran Associates, Inc., 2980–2988.
- [6] Cyrus DiCiccio, Sriram Vasudevan, Kinjal Basu, Krishnaram Kenthapadi, and Deepak Agarwal. 2020. Evaluating Fairness Using Permutation Tests. In *KDD*. ACM, 1467–1477.
- [7] Chen Gao, Yu Zheng, Wenjie Wang, Fuli Feng, Xiangnan He, and Yong Li. 2022. Causal inference in recommender systems: A survey and future directions. *arXiv:2208.12397* (2022).
- [8] Yingqiang Ge, Shuya Zhao, Honglu Zhou, Changhua Pei, Fei Sun, Wenwu Ou, and Yongfeng Zhang. 2020. Understanding Echo Chambers in E-Commerce Recommender Systems. In *SIGIR*. ACM, 2261–2270.
- [9] Ruining He and Julian McAuley. 2016. Fusing Similarity Models With Markov Chains for Sparse Sequential Recommendation. In *ICDM*. IEEE, 191–200.
- [10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.
- [11] Yue He, Peng Cui, Zheyang Shen, Renzhe Xu, Furui Liu, and Yong Jiang. 2021. Daring: Differentiable causal discovery with residual independence. In *KDD*. 596–605.
- [12] Yue He, Zimu Wang, Peng Cui, Hao Zou, Yafeng Zhang, Qiang Cui, and Yong Jiang. 2022. CausPref: Causal Preference Learning for Out-of-Distribution Recommendation. In *WWW*. ACM, 410–421.
- [13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*.
- [14] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *ICLR*.
- [15] Amir H. Jadidinejad, Craig Macdonald, and Iadh Ounis. 2021. The Simpson’s Paradox in the Offline Evaluation of Recommendation Systems. *TOIS* 40, 1 (2021).
- [16] Hao Jiang, Wenjie Wang, Yinwei Wei, Zan Gao, Yinglong Wang, and Liqiang Nie. 2020. What Aspect Do You Like: Multi-Scale Time-Aware User Interest Modeling for Micro-Video Recommendation. In *MM*. ACM, 3487–3495.
- [17] Thorsten Joachims, Adith Swaminathan, and Tobias Schnabel. 2017. Unbiased Learning-to-Rank with Biased Feedback. In *WSDM*. ACM, 781–789.
- [18] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. IEEE Computer Society, 197–206.
- [19] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *ICLR*.
- [20] Masanori Koyama and Shoichiro Yamaguchi. 2021. When is invariance useful in an Out-of-Distribution Generalization problem ? (2021).
- [21] Lai, Siqu and Meng, Erli and Zhang, Fan and Li, Chenliang and Wang, Bin and Sun, Aixun. 2022. An Attribute-Driven Mirror Graph Network for Session-based Recommendation. In *SIGIR*. ACM, 1674–1683.

- [22] Latifi, Sara and Mauro, Noemi and Jannach, Dietmar. 2021. Session-aware recommendation: A surprising quest for the state-of-the-art. *Information Sciences* 573 (2021), 291–315.
- [23] Xiaopeng Li and James She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *KDD*. ACM, 305–314.
- [24] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. 2021. Towards Personalized Fairness Based on Causal Notion. In *SIGIR*. ACM, 1054–1063.
- [25] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational Autoencoders for Collaborative Filtering. In *WWW*. ACM, 689–698.
- [26] Xixun Lin, Jia Wu, Chuan Zhou, Shirui Pan, Yanan Cao, and Bin Wang. 2021. Task-adaptive Neural Process for User Cold-Start Recommendation. In *WWW*. ACM, 1306–1316.
- [27] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyang Shen. 2021. Heterogeneous Risk Minimization. In *ICML*. PMLR, 6804–6814.
- [28] Liu, Dugang and Cheng, Pengxiang and Zhu, Hong and Dong, Zhenhua and He, Xiuqiang and Pan, Wei and Ming, Zhong. 2021. Mitigating Confounding Bias in Recommendation via Information Bottleneck. In *RecSys*. ACM, 351–360.
- [29] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. 2020. Weakly-supervised Disentanglement Without Compromises. In *ICML*. PMLR, 6348–6359.
- [30] Huishi Luo, Fuzhen Zhuang, Ruobing Xie, Hengshu Zhu, and Deqing Wang. 2023. A Survey on Causal Inference for Recommendation. *arXiv:2303.11666* (2023).
- [31] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning Disentangled Representations for Recommendation. In *NeurIPS*. Curran Associates, Inc., 5712–5723.
- [32] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled Self-supervision in Sequential Recommenders. In *KDD*. ACM, 483–491.
- [33] Jarana Manotumruksa and Emine Yilmaz. 2020. Sequential-based Adversarial Optimisation for Personalised Top-n Item Recommendation. In *SIGIR*. ACM, 2045–2048.
- [34] Judea Pearl. 2009. *Causality*. Cambridge university press.
- [35] Judea Pearl and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc.
- [36] Massimo Quadroni, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing Session-based Recommendations With Hierarchical Recurrent Neural Networks. In *RecSys*. ACM, 130–137.
- [37] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *UAI*. AUAI Press, 452–461.
- [38] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing Personalized Markov Chains for Next-basket Recommendation. In *WWW*. ACM, 811–820.
- [39] Donald B Rubin. 2005. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *JASA* 100, 469 (2005), 322–331.
- [40] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing Between Capsules. In *NeurIPS*, Vol. 30.
- [41] Naveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential variational autoencoders for collaborative filtering. In *WSDM*. ACM, 600–608.
- [42] Yuta Saito, Suguru Yaginuma, Yuta Nishino, Hayato Sakata, and Kazuhide Nakata. 2020. Unbiased Recommender Learning from Missing-Not-At-Random Implicit Feedback. In *WSDM*. ACM, 501–509.
- [43] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *WWW*. ACM, 285–295.
- [44] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward Causal Representation Learning. *Proc. IEEE* 109, 5 (2021), 612–634.
- [45] Xiang-Rong Sheng, Liqin Zhao, Guorui Zhou, Xinyao Ding, Binding Dai, Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, et al. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi-Domain CTR Prediction. In *CIKM*. ACM, 4104–4113.
- [46] Zihua Si, Xueran Han, Xiao Zhang, Jun Xu, Yue Yin, Yang Song, and Ji-Rong Wen. 2022. A Model-Agnostic Causal Learning Framework for Recommendation using Search Data. In *WWW*. ACM.
- [47] Nakarin Sritrakool and Saranya Maneeroj. 2021. Personalized Preference Drift Aware Sequential Recommender System. *IEEE Access* 9 (2021), 155491–155506.
- [48] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. ACM, 1441–1450.
- [49] Ke Sun, Tiejun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *AAAI*, Vol. 34. 214–221.
- [50] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *WSDM*. ACM, 565–573.
- [51] Khanh Hiep Tran, Azin Ghazimatin, and Rishiraj Saha Roy. 2021. Counterfactual Explanations for Neural Recommenders. In *SIGIR*. ACM, 1627–1631.



- [52] Cong Wang, Yifeng Zheng, Jinghua Jiang, and Kui Ren. 2018. Toward privacy-preserving personalized recommendation services. *Engineering* 4, 1 (2018), 21–28.
- [53] Chao Wang, Hengshu Zhu, Peng Wang, Chen Zhu, Xi Zhang, Enhong Chen, and Hui Xiong. 2021. Personalized and Explainable Employee Training Course Recommendations: A Bayesian Variational Approach. *TOIS* 40, 4 (2021).
- [54] Pengfei Wang, Yu Fan, Long Xia, Wayne Xin Zhao, ShaoZhang Niu, and Jimmy Huang. 2020. KERL: A knowledge-guided reinforcement learning model for sequential recommendation. In *SIGIR*. 209–218.
- [55] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded Recommendation for Alleviating Bias Amplification. In *KDD*. ACM, 1717–1725.
- [56] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Click can be Cheating: Counterfactual Recommendation for Mitigating Clickbait Issue. In *SIGIR*. ACM, 1288–1297.
- [57] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*. ACM, 3562–3571.
- [58] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *SIGIR*. ACM, 1001–1010.
- [59] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2019. Doubly Robust Joint Learning for Recommendation on Data Missing Not at Random. In *ICML*. PMLR, 6638–6647.
- [60] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM*. ACM, 1605–1614.
- [61] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, Xu Chen, Yongfeng Zhang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Counterfactual Data-augmented Sequential Recommendation. In *SIGIR*. ACM, 347–356.
- [62] Charinya Wangwatcharakul and Sartra Wongthanavasu. 2021. A novel temporal recommender system based on multiple transitions in user preference drift and topic review evolution. *Expert Systems with Applications* 185 (2021), 115626.
- [63] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A context-aware user-item representation learning for item recommendation. *TOIS* 37, 2 (2019), 1–29.
- [64] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *AAAI*. AAAI Press, 346–353.
- [65] Wu, Peng and Li, Haoxuan and Deng, Yuhao and Hu, Wenjie and Dai, Quanyu and Dong, Zhenhua and Sun, Jie and Zhang, Rui and Zhou, Xiao-Hua. 2022. On the opportunity of causal learning in recommendation systems: Foundation, estimation, prediction and challenges. In *IJCAI*. 23–29.
- [66] Lianghao Xia, Chao Huang, Yong Xu, Huance Xu, Xiang Li, and Weiguo Zhang. 2021. Collaborative Reflection-Augmented Autoencoder Network for Recommender Systems. *TOIS* 40, 1 (2021).
- [67] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *WWW*. ACM, 449–459.
- [68] 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* (2023).
- [69] Xu, Shuyuan and Ge, Yingqiang and Li, Yunqi and Fu, Zuohui and Chen, Xu and Zhang, Yongfeng. 2021. Causal collaborative filtering. (2021).
- [70] Yutaro Yamada, Ofir Lindenbaum, Sahand Negahban, and Yuval Kluger. 2020. Feature selection using stochastic gates. In *ICML*. PMLR, 10648–10659.
- [71] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. 2021. CausalVAE: Disentangled Representation Learning via Neural Structural Causal Models. In *CVPR*. IEEE, 9593–9602.
- [72] Hongzhi Yin, Xiaofang Zhou, Bin Cui, Hao Wang, Kai Zheng, and Quoc Viet Hung Nguyen. 2016. Adapting to user interest drift for poi recommendation. *IEEE Transactions on Knowledge and Data Engineering* 28, 10 (2016), 2566–2581.
- [73] Runsheng Yu, Yu Gong, Xu He, Yu Zhu, Qingwen Liu, Wenwu Ou, and Bo An. 2021. Personalized Adaptive Meta Learning for Cold-start User Preference Prediction. In *AAAI*. AAAI Press, 10772–10780.
- [74] Fajie Yuan, Xiangnan He, Alexandros Karatzoglou, and Liguang Zhang. 2020. Parameter-efficient Transfer from Sequential Behaviors for User Modeling and Recommendation. In *SIGIR*. ACM, 1469–1478.
- [75] Farhad Zafari, Irene Moser, and Tim Baarslag. 2019. Modelling and analysis of temporal preference drifts using a component-based factorised latent approach. *Expert Systems with Applications* 116 (2019), 186–208.
- [76] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. CauseRec: Counterfactual User Sequence Synthesis for Sequential Recommendation. In *SIGIR*. ACM, 367–377.
- [77] Wenhao Zhang, Wentian Bao, Xiao-Yang Liu, Keping Yang, Quan Lin, Hong Wen, and Ramin Ramezani. 2020. Large-Scale Causal Approaches to Debiasing Post-Click Conversion Rate Estimation with Multi-Task Learning. In *WWW*. ACM, 2775–2781.
- [78] Xingxuan Zhang, Peng Cui, Renzhe Xu, Linjun Zhou, Yue He, and Zheyang Shen. 2021. Deep stable learning for out-of-distribution generalization. In *CVPR*. IEEE, 5372–5382.

- [79] Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal Intervention for Leveraging Popularity Bias in Recommendation. In *SIGIR*. ACM, 11–20.
- [80] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-domain recommendation via preference propagation graphnet. In *CIKM*. ACM, 2165–2168.
- [81] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-domain recommendation for cold-start users via aspect transfer network. In *SIGIR*. ACM, 229–238.
- [82] Zhou, Guorui and Mou, Na and Fan, Ying and Pi, Qi and Bian, Weijie and Zhou, Chang and Zhu, Xiaoqiang and Gai, Kun. 2019. Deep interest evolution network for click-through rate prediction. In *AAAI*, Vol. 33. 5941–5948.
- [83] Zhou, Guorui and Zhu, Xiaoqiang and Song, Chenru and Fan, Ying and Zhu, Han and Ma, Xiao and Yan, Yanghui and Jin, Junqi and Li, Han and Gai, Kun. 2018. Deep interest network for click-through rate prediction. In *KDD*. ACM, 1059–1068.
- [84] Nengjun Zhu, Jian Cao, Xinjiang Lu, and Hui Xiong. 2021. Learning a Hierarchical Intent Model for Next-Item Recommendation. *TOIS* 40, 2 (2021).
- [85] Yaochen Zhu, Jing Ma, and Jundong Li. 2023. Causal Inference in Recommender Systems: A Survey of Strategies for Bias Mitigation, Explanation, and Generalization. *arXiv:2301.00910* (2023).
- [86] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized Transfer of User Preferences for Cross-domain Recommendation. In *WSDM*. ACM.
- [87] Hao Zou, Peng Cui, Bo Li, Zheyang Shen, Jianxin Ma, Hongxia Yang, and Yue He. 2020. Counterfactual Prediction for Bundle Treatment. In *NeurIPS*. 19705–19715.