Personalized Latent Structure Learning for Recommendation

Shengyu Zhang, Fuli Feng, Kun Kuang*, Wenqiao Zhang, Zhou Zhao, Hongxia Yang, Tat-Seng Chua, Fei Wu*, *Senior Member, IEEE*

Abstract—In recommender systems, users' behavior data are driven by the interactions of user-item latent factors. To improve recommendation effectiveness and robustness, recent advances focus on latent factor disentanglement via variational inference. Despite significant progress, uncovering the underlying interactions, *i.e.*, dependencies of latent factors, remains largely neglected by the literature. To bridge the gap, we investigate the joint disentanglement of user-item latent factors and the dependencies between them, namely latent structure learning. We propose to analyze the problem from the causal perspective, where a latent structure should ideally reproduce observational interaction data, and satisfy the structure acyclicity and dependency constraints, *i.e.*, causal prerequisites. We further identify the recommendation-specific challenges for latent structure learning, *i.e.*, the subjective nature of users' minds and the inaccessibility of private/sensitive user factors causing universally learned latent structure to be suboptimal for individuals. To address these challenges, we propose the personalized latent structure learning framework for recommendation, namely PlanRec, which incorporates 1) differentiable Reconstruction, Dependency, and Acyclicity regularizations to satisfy the causal prerequisites; 2) Personalized Structure Learning (PSL) which personalizes the universally learned dependencies through probabilistic modeling; and 3) uncertainty estimation which explicitly measures the uncertainty of structure personalization, and adaptively balances personalization and shared knowledge for different users. We conduct extensive experiments on two public benchmark datasets from MovieLens and Amazon, and a large-scale industrial dataset from Alipay. Empirical studies validate that PlanRec discovers effective shared/personalized structures, and successfully balances shared knowledge and personalization via rational uncertainty estimation.

Index Terms—Latent Structure Learning, Structure Personalization, Uncertainty Estimation, Recommender Systems

1 Introduction

In recommender systems, users' behaviors are driven by the interactions of user-item latent factors [1]. A latent factor implies a particular user characteristic (*e.g.*, user career) or item aspect (*e.g.*, item category). For example, in an e-commerce platform, we might discover that young men working in IT love buying electronic products and can afford high-priced products. The buying behavior is driven by a combination of latent factors including, but are not limited to, {Age, Career, Category, Price}.

Recently, there has been increasing research interest in learning disentangled user representations, based chiefly on user behavior data, for recommendation [2], [3], [4], [5], [6], [7]. In the literature of recommendation, disentanglement [1] typically refers to uncovering multiple distinctive latent factors of users/items, and representing them into semantic dense vectors. For instance, Wang *et al*. [3] claim that explicit latent factor disentanglement can improve recommendation robustness and interpretability. This work and many others [5], [8], [9], [10] exploit Variational Auto-encoders (VAE) for disentanglement, where the reconstruction objective and the KL regularization on the posterior [11] jointly learn effective latent factors [12]. More recently, ACVAE [10] improves the disentanglement distinction via adversarial learning and contrastive learning. DESTINE [13] decomposes the feature interaction of user-item features into terms for pairwise and unary semantics. Besides user behavior data, some works explore auxiliary content information, such as vision-language data, for improved disentanglement [3], [6], [14].

Despite their notable success, existing disentangled recommendation frameworks generally neglect the dependencies between user-item latent factors. Intuitively, how users' factors cause items' factors reflect the decision-making patterns of users, *e.g.*, $\{$men $(Gender) \rightarrow$ electronic products $(Category)$, IT $(Career) \rightarrow price, \dots\}$. Further disentangling such decision-making patterns might potentially improve the recommendation performance and robustness. Hence, it is necessary to disentangle both the user-item latent factors and the dependencies of these factors in recommendation (*cf.*, Figure 1).

In this work, we refer to such a joint disentanglement as ***latent structure*** learning in recommendation. Latent structure learning aims to uncover and represent the latent factors of both users and items, and simultaneously learn a dependency matrix of these latent factors. Each element in the dependency matrix represents the effect of the parent latent factor on the child latent factor. In technique, these

- *Shengyu Zhang, Kun Kuang and Zhou Zhao are with College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China. E-mail: {sy_zhang, kunkuang, zhouzhao}@zju.edu.cn*
- *Fei Wu is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China, with the Shanghai Institute for Advanced Study, Zhejiang University, Shanghai 201210, China, and also with the Shanghai AI Laboratory, Shanghai 200232, China. E-mail: wufei@zju.edu.cn*
- *Fuli Feng is with the University of Science and Technology of China, Hefei 230026, China. E-mail: fulifeng93@gmail.com*
- *Wenqiao Zhang and Tat-Seng Chua are with the National University of Singapore, Singapore. E-mail: wenqiao@nus.edu.sg, chuats@comp.nus.edu.sg*
- *Hongxia Yang is with Alibaba Group, Hangzhou 310052, China. E-mail: yang.yhx@alibaba-inc.com*

* *Corresponding authors*

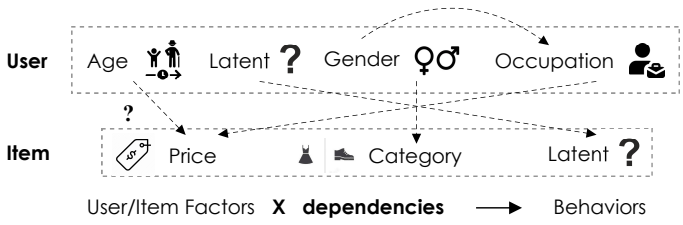

Fig. 1: An illustration of latent structure learning for recommendation, *i.e.*, disentangling user-item latent factors and the dependencies of factors.

latent factors and the dependency matrix represent a directed acyclic graph, which we denote as the latent structure of recommendation. To improve the dependency discovery of user-item latent factors, we resort to some key ideas in causal discovery [15] (also known as causal structure learning [16], [17], [18], or causal induction [19] in causal theory), and propose three regularizations that the latent structure should satisfy:

- **Reconstruction Regularization**, which indicates that user-item latent factors together with their dependencies (decision-making patterns) should be able to reproduce the historical behaviors of users. We technically achieve the regularization by adopting a decoder that takes latent factors and dependencies as input and learns to reconstruct the original user interaction sequence.
- **Dependency Regularization**, which indicates that the value of a latent factor is determined when the values of parent latent factors are determined. Intuitively, we can estimate the preferred item price range when a user's price-sensitive factors (*e.g.*, career) are given. In technique, we learn to predict the representation of each latent factor based on the representations of its parent factors indicated by the dependency matrix.
- **Acyclicity Regularization**, which ensures that there is no cycle in the latent structure; otherwise, each latent factor can influence and can be influenced by all other factors in the cycle indirectly, which is less practical. For example, the preferred item categories might progressively change according to the increase of a user's age, but not vice versa. We employ a differentiable constraint function to penalize structural cycles in the latent structure.

As such, we could build a **base** latent structure learning model. Wherein a vanilla VAE model achieves latent factor disentanglement with user-item interaction data as input. These latent factors together with the trainable dependency matrix are learned to satisfy the above three regularizations and the vanilla recommendation objective. After training, the latent structure includes the overall dependencies of user-item factors shared by the majority of users, namely the shared latent structure (**SLS**). Intuitively, SLS reflects users' overall decision-making patterns in a recommender system, such as $Career \rightarrow Price$ in an e-commerce platform.

However, this base latent structure learning method has some drawbacks. Due to the subjective nature of users' minds and the inaccessibility of many sensitive/private user factors [20], the universally learned latent structure might not reflect the decision-making patterns of some users. For example, we might discover that, for most users, the Career factor causes the Price factor of items that these users will interact with, *i.e.*, ($Career \rightarrow Price$). However, some users might not care about the price ($Career \not\rightarrow Price$) because they are from wealthy families. The factor of family wealth is sensitive/private and can hardly be disentangled from behavior data and captured in the latent structure. Unmeasured factors are arguably causing varying responses of individuals given the same treatment [20]. Moreover, users' minds are subjective and some users might not follow the crowd. Therefore, it is necessary to explore the personalized latent structure for each user, where the latent factors and the dependencies reflect the individual decision-making patterns, as opposed to the shared latent structure.

In this work, we refer to the user-wise disentanglement as ***personalized*** *latent structure learning*. The major challenge lies in the sparse and noisy nature of users' behaviors, driving structure personalization challenging for some users, especially for users with low activeness (fewer behaviors) or vague behavior patterns (more noises), leading to underfitting or overfitting issues of personalized structure learning. Therefore, it can be necessary to accommodate imperfect personalization in personalized latent structure learning.

To address the above challenges, we propose a ***Pe***rsonalized ***La***tent Structure Lear***n***ing framework for recommendation, namely **PlanRec**. In essence, PlanRec introduces *latent structure personalization* and *balancing personalization and shared knowledge* for each individual over the base latent structure learning model:

- **Latent structure personalization**. Specifically, upon the shared latent structure (SLS) learned by the base model, PlanRec estimates the user-wise masking probabilities for each latent factor and each element in the dependency matrix, where a masking probability reflects to what extent the user's decision-making does not rely on the corresponding dependency or latent factor. Then, we could obtain personalized latent structure (PLS) for each user via probabilistic masking.
- **Balancing personalization and shared knowledge**. Moreover, thanks to the probabilistic modeling of personalized latent structure, we can mimic the Bayesian Neural Network (BNN) for personalization uncertainty estimation. Technically, we can estimate the expectation and variance of predictions based on multiple randomly sampled PLSs. The variance of PLS predictions is treated as the uncertainty of the latent structure personalization, eventually balancing the predictions from SLS and PLS. The motivation is intuitive – when we are more uncertain about the current user's decision-making patterns (PLS), we trust more the decision-making patterns learned from all users (SLS) in the system.

To summarize, this work makes the following key contributions:

- We analyze how recommender systems can benefit from latent structure learning, and propose three regularizations for learning the shared latent structure.
- The proposed PlanRec learns personalized latent structures via user-conditioned probabilistic masking.
- PlanRec achieves a balance of the predictions from SLS and PLS by uncertainty estimation, thus accommodat-

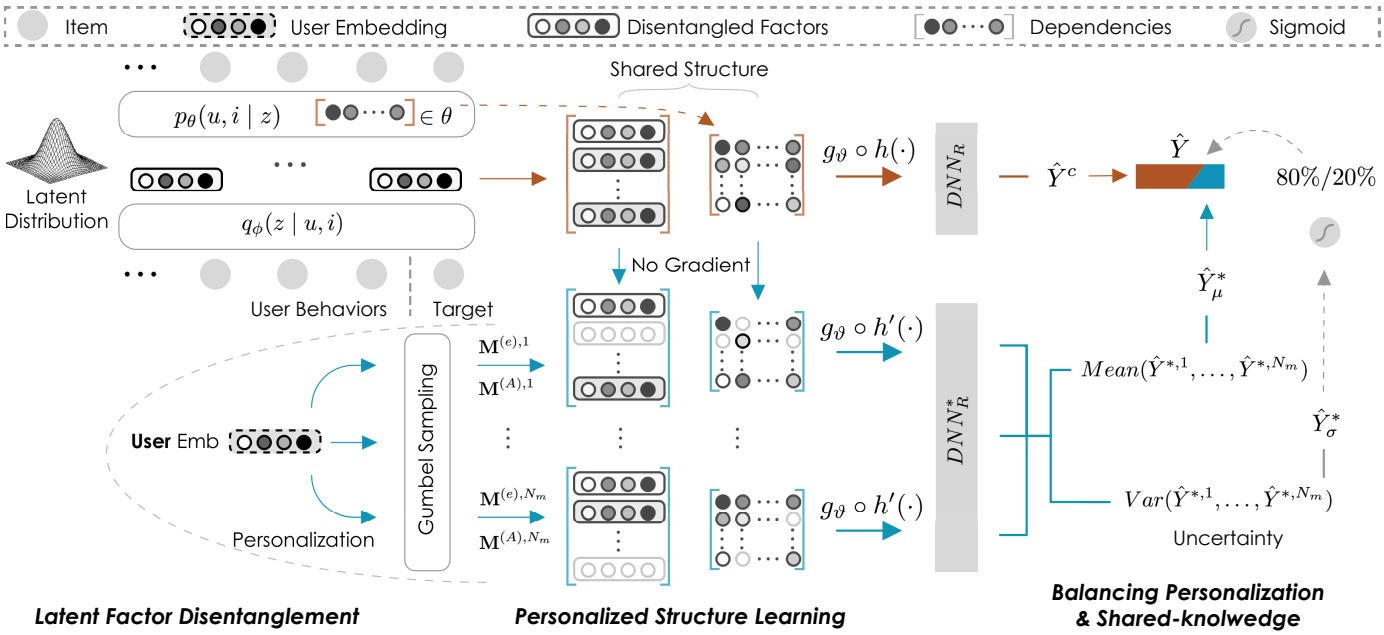

Fig. 2: Schematic of the proposed PlanRec framework, which consists of three critical components: 1) shared structure learning that learns the shared dependencies of latent factors discovered by VAE based on three hypotheses (omitted here for simplicity); 2) personalized structure learning that probabilistically sample personalized structures conditioned on the shared structure and user embedding; 3) uncertainty estimation that measures the personalization uncertainty, and accordingly balance the predictions of personalization and shared knowledge.

ing imperfect personalization.
- We conduct extensive experiments on three real-world datasets. The insightful analysis demonstrates the rationality and effectiveness of PlanRec.

2 METHODS

2.1 Problem Formulation.

Recommendation. Recommendation aims to learn a score function $f_\Theta(u,i)$ parameterized by Θ that evaluates the probability of user u interacting with item i. The parameters Θ are typically learned from users' historical behavior data $\mathcal{D} = \{(u,i,Y_{u,i})\}$ where $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $Y_{u,i} \in \{0,1\}$. $Y_{u,i}$ denotes whether user u has been observed to interact with item i. Generally, a recommendation loss $\mathcal{L}_R$ can be defined as:

$$\mathcal{L}_R = \sum\nolimits_{(u,i,Y_{u,i})\in\mathcal{D}} l(f_\Theta(u,i), Y_{u,i}), \tag{1}$$

where $l(\cdot)$ denotes the loss function such as cross entropy [21]. During inference, for each user, all candidate items will be ranked according to the preference scores obtained by $f_\Theta(u,i)$. Top-ranked items will be recommended to users.

Latent Structure Learning. In this work, latent structure learning is defined as uncovering the latent factors of both users and items, *i.e.*, $z_1, z_2, \ldots, z_N$, as well as the dependencies in between, *i.e.*, $\mathbf{A} \in \mathbb{R}^{N*N}$ from the observational interaction data $\mathcal{D}$. Each latent factor z represents a particular characteristic of a user or an item (*e.g.*, user career, or item price). We use the bold letter $\mathbf{z}$ to denote the vectorial representation of z. Each element $A_{i,j} \in \{0,1\}$ in $\mathbf{A}$ represents whether the i-th latent factor z_i has a direct effect on the j-th latent factor z_j. In this work, we consider two kinds of latent structure learning, *i.e.*, shared structure learning and personalized structure learning, which aim to uncover the *shared* latent structure of all users and the *personalized* latent structure for each individual, respectively.

2.2 Model Overview

The proposed PlanRec consists of four critical components:
- **Latent Factor Disentanglement** (*cf.* Section 2.3), which decomposes the user-item interaction data into latent factors via variational inference, following existing disentangled recommendation techniques.
- **Shared Structure Learning** (*cf.* Section 2.4), which introduces a trainable matrix representing the dependencies between latent factors. The disentangled latent factors and the dependency matrix, *i.e.*, the latent structure, are learned to satisfy the reconstruction, dependency, and acyclicity regularizations.
- **Personalized Structure Learning** (*cf.* Section 2.5), which personalizes the latent structure for each individual with probabilistic masking.
- **Balancing Personalization and Shared Knowledge** (*cf.* Section 2.6), which estimates the prediction variance of multiple probabilistically sampled PLSs as the personalization uncertainty, serving as the basis for balancing the predictions based on SLS and PLSs.

The schematic of PlanRec is shown in Figure 2.

2.3 Latent Factor Disentanglement

Latent factor disentanglement aims to decompose the observational interaction data into semantically different dense

vectors, where each vector represents a particular user/item aspect. We follow the Variational Auto-encoder (VAE) paradigm for latent factor disentanglement. VAE assumes that observational data samples are generated based on unobservable latent factors. For recommendation, the latent factors come from the user side (*e.g.*, gender, age, career) and the target item side (*e.g.*, price, and category). Formally, considering the following generative model and the posterior density:

$$p_\theta(u,i) = \int p_\theta(u,i \mid z) p_\theta(z) dz, \tag{2}$$

$$p_\theta(z \mid u,i) = p_\theta(u,i \mid z) p_\theta(z) / p_\theta(u,i), \tag{3}$$

where z denotes the disentangled latent factors of users and items. Both the integral of marginal likelihood $p_\theta(u,i \mid z)$ and the posterior density $p_\theta(z \mid u,i)$ are intractable. As such, a high-capacity recognition model $q_\phi(z \mid u,i)$ parameterized by ϕ is introduced to approximate the intractable posterior $p_\theta(z \mid u,i)$. Kullback–Leibler divergence is leveraged for such distribution approximation:

$$\begin{aligned} &KL\left(q_\phi(z \mid u,i) \,||\, p_\theta(z \mid u,i)\right) \\ &= -\sum_z q_\phi(z \mid u,i) \left[\log\left(\frac{p_\theta(u,i,z)}{q_\phi(z \mid u,i)}\right) - \log(p_\theta(u,i))\right]. \end{aligned} \tag{4}$$

With simple arrangement, we have:

$$\begin{aligned} \log(p_\theta(u,i)) &= KL\left(q_\phi(z \mid u,i) \,||\, p_\theta(z \mid u,i)\right) + ELBO \\ &>= ELBO, \end{aligned} \tag{5}$$

where the KL term is non-negative. As such, maximizing the evidence lower bound $ELBO$ approximates maximizing the joint probability of observed user-item pairs (u,i).

$$\begin{aligned} ELBO &= \sum_z q_\phi(z \mid u,i) \log\left(\frac{p_\theta(u,i \mid z) p_\theta(z)}{q_\phi(z \mid u,i)}\right) \\ &= \sum_z q_\phi(z \mid u,i) \left[\log(p_\theta(u,i \mid z)) + \log\left(\frac{p_\theta(z)}{q_\phi(z \mid u,i)}\right)\right] \\ &= \underbrace{E_{q_\phi}\left[\log(p_\theta(u,i \mid z))\right]}_{\text{Reconstruction}} - \underbrace{KL(q_\phi(z \mid u,i) \,||\, p_\theta(z))}_{\text{Regularization}}. \end{aligned} \tag{6}$$

Implementing $q_\phi(z \mid u,i)$. We implement $q_\phi(z \mid u,i)$ as a neural network based encoder which transforms input user/item features into latent factors. We propose to decompose $q_\phi(z \mid u,i)$ into two components where the first component can be any neural architectures that evolve interactions of the user and the target item, including (but are not limited to) sequential recommendation models [22] and Factorization Machine based models [23]. The second component is a multi-layer perceptron followed by a reshape operation, which transforms the output of the first component into multiple vectors $[\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_N]$ where each of them represents a particular user/item latent factor and N is the number of all latent factors.

Implementing $p_\theta(u,i \mid z)$ **and** $p_\theta(z)$. $p_\theta(u,i \mid z)$ is a decoder which reconstructs the user/item input based on the latent factors z. We use multi-layer perceptron for decoding. We leverage cosine distance to measure the difference between decoded representations and input user/item embeddings. The distance serves as the reconstruction loss. To achieve disentanglement of latent factors, the prior $p_\theta(z)$ should be factorized, and is then set to $\mathcal{N}\left(\mathbf{0}, b_0^2\mathbf{I}\right)$.

Disentanglement with Prior. Given user-item attributes as the prior knowledge, we could further enhance the disentanglement by building one-to-one correspondence between disentangled latent factors and physical user-item attributes. Technically, we adopt multi-task learning, where each attribute-specific predictor takes the corresponding latent factor as input and learns to predict the attribute value. We adopt classification for attributes with unordered values (*e.g.*, user gender), and regression for attributes with ordered values (*e.g.*, item price) for training the predictors.

2.4 Shared Structure Learning

Shared structure learning aims to estimate the dependency values that should apply for *most users* among disentangled latent factors. Understanding the dependencies from user latent factors to item latent factors (*e.g.*, User $Career \xrightarrow{?}$ Item $Price$) could better uncover the decision-making patterns of users for recommendation. We introduce a Structural Causal Model (SCM) [24] layer to model the dependencies. An SCM takes exogenous factors (*e.g.*, sampled from multivariate Gaussian) as input, and outputs endogenous factors, which encode the shared dependencies $\mathbf{A}$. Formally, we have:

$$\begin{aligned} \mathbf{e} &= \mathbf{A}^T\mathbf{e} + \mathbf{z} = (I - \mathbf{A}^T)^{-1}\mathbf{z} \\ &= h(\mathbf{z}; \mathbf{A}), \quad \mathbf{z} \sim q_\phi(z \mid u,i), \end{aligned} \tag{7}$$

where $\mathbf{z}$ is sampled from $q_\phi(z \mid u,i)$, which is regularized to approach $\mathcal{N}\left(\mathbf{0}, b_0^2\mathbf{I}\right)$ during training. At inference, we take the mode of $q_\phi(z \mid u,i)$ as the representation of z, *i.e.*, $\mathbf{z}$. $\mathbf{e}$ denotes the endogenous latent factors. $\mathbf{A}$ denotes the trainable matrix where each element represents the dependency between a particular pair of latent factors. Equation (7) is a standard linear structural equation model. In practice, we introduce non-linearity to the generation process of latent factors through a neural network $g_\vartheta(\cdot)$:

$$\hat{\mathbf{e}}_i = g_\vartheta(\mathbf{A}_i * \mathbf{e}), \tag{8}$$

where $*$ denotes the Hadamard Product. $\mathbf{A}_i$ denotes the coefficients of all other latent factors to the ith factor. The underlying intuition of this operation is to recover the ith factor from its parent factors while other factors are masked as zero values. Final recommendation predictions are made based on $\hat{\mathbf{e}}$, which encodes latent factors and their dependencies with non-linear transformations. Parameters $\mathbf{A}$ and ϑ will be updated in back-propagation. However, the recommendation objective does not necessarily guarantee that $\mathbf{A}$ learns the dependencies of latent factors. To improve dependency identification, we resort to causal discovery and propose three hypotheses (*cf.* Section 1) and three regularization objectives for training $\mathbf{A}$ as follows:

Reconstruction Regularization . User-item latent factors and their dependencies constitute the latent structure, reflecting the decision-making patterns of users. With the latent structure, we should be able to recover the historical behaviors. In this regard, we include parameter $\mathbf{A}$ in the parameter set θ in the reconstruction of historical behaviors $p(u,i \mid z)$, *cf.* Equation (6). We transform the exogenous

factors $\mathbf{z}$ into endogenous factors $\hat{\mathbf{e}}_i$ with Equation (7) - Equation (8) before reconstruction.

Dependency Regularization . Parent factors that cause a particular factor should be able to reproduce the factor. In this sense, we introduce a structural prediction loss to enhance the dependencies between latent factors:

$$\mathcal{L}_{struct} = \left\|\hat{\mathbf{e}} - \mathbf{e}\right\|^2. \tag{9}$$

Acyclicity Regularization . A cycle in the latent structure means that any factor in the cycle could be indirectly influenced by all other factors in the cycle, which is less practical. For example, user age/gender causes the preferred item categories but not vice versa. To enhance acyclicity in the latent structure, we borrow the differentiable regularization from [17] as follows:

$$\mathcal{L}_{dag} = \operatorname{tr}\left((\mathbf{I} + \alpha\mathbf{A} * \mathbf{A})^n\right) - n, \tag{10}$$

where tr denotes the matrix trace and α is a hyper-parameter which depends on a prior estimation of the largest eigenvalue of $\mathbf{A} * \mathbf{A}$.

2.5 Personalized Structure Learning

In the shared structure learning, the dependency matrix $\mathbf{A}$ inside the latent structure is learned from all users' data and shared by all users in model inference. Therefore, $\mathbf{A}$ represents the overall knowledge shared by all users (*e.g.*, User $Career \rightarrow$ Item $Price$). However, in recommender systems, users' behaviors are highly personalized and subjective. A shared latent structure might not accommodate such personalization. In addition, the disentanglement and shared structure learning might hardly capture sensitive/private latent factors (*e.g.*, users being from wealthy families). Existing works from other domains find that unmeasured factors could result in varying responses given the same treatment [20], which implies that the effect (dependency) of the parent latent factor on the child latent factor could vary across different users. For example, patients may show different responses to the same medical treatment due to unmeasured nutrition and health status [20]. Therefore, the universally learned dependencies of some factors might not be optimal for some users (*e.g.*, User $Career \not\rightarrow$ Item $Price$). It is necessary to perform structure personalization to estimate the personalized dependencies for each user, *i.e.*, personalized structure learning.

To prevent catastrophic deviation from the shared structure learned by all users (larger variances), we do not generate a personalized structure *from scratch* for each user. Instead, we propose to adopt a personalized mask applied to both the latent factors and the shared structure. We first obtain the masking confidence scores for latent factors and shared structure conditioned on the user embedding:

$$\bar{\mathbf{M}}^{(A)}, \bar{\mathbf{M}}^{(e)} = MLP_{mask}(\mathbf{u}), \tag{11}$$

where each element in $\bar{\mathbf{M}}^{(e)}$ or $\bar{\mathbf{M}}^{(A)}$ is a confidence score indicating to what extent the user does not care about the corresponding latent factor or shared dependencies when he/she makes behavior decisions, *e.g.*, User $Career \not\rightarrow$ Item $Price$ due to being from a wealthy family. $\mathbf{u}$ denotes the embedding of user u. Then, we leverage Gumbel sampling [25] based on these confidence scores. After sampling, each element $\bar{m}_i \in \bar{\mathbf{M}}^{(e)}$ will be transformed to:

$$m_i^{(e)} = \frac{\exp\left(\left(\log\left(\bar{m}_i^{(e)}\right) - \log(-\log g_i)\right)/\tau\right)}{\sum_{m_j^{(e)} \in \bar{\mathbf{M}}^{(e)}} \exp\left(\left(\log\left(\bar{m}_j^{(e)}\right) - \log(-\log g_j)\right)/\tau\right)}, \tag{12}$$

where $m_i^{(e)}$ is the ith element of the sampled mask $\mathbf{M}^{(e)}$ and $g \sim \text{Uniform}(0, 1)$. $\mathbf{M}^{(A)}$ is obtained similarly. τ controls the smoothness of the resulted distribution. Note that after Gumbel softmax with $\tau > 0$, the elements in the mask $\mathbf{M}^{(A)}$ are still probabilities. To have $0/1$ mask with back-propagation, we adopt the Straight-through trick [26]. Specifically, in the forward pass, the Top-K elements with the highest probabilities will become one while the remaining elements will become zero. In the back-propagation, actual values (probabilities) of elements in the mask $\mathbf{M}^{(A)}$ will be used, approximating the actual gradients. The generation process of the factor mask $\mathbf{M}^{(e)}$ shares similar spirits. Upon $\mathbf{M}^{(A)}$ and $\mathbf{M}^{(e)}$, we obtain the personalized latent factors:

$$\begin{aligned} \mathbf{e}^* &= h'(\mathbf{z}, \mathbf{M}^{(A)}, \mathbf{M}^{(e)}; \mathbf{A}), \quad \mathbf{z} \sim q_\phi(z \mid u, i), \\ &= (I - (\mathbf{M}^{(A)} * \mathbf{A}^T)^{-1})(\mathbf{M}^{(e)} * \mathbf{z}), \end{aligned} \tag{13}$$

$$\hat{\mathbf{e}}_i^* = g_\vartheta((\mathbf{M}_i^{(A)} * \mathbf{A}_i) * (\mathbf{M}^{(e)} * \mathbf{e}^*)), \tag{14}$$

where $*$ denotes the Hadamard Product. It is noteworthy that we block the back-propagation on $\mathbf{A}$ when computing Equation (13)-(14). This operation shields the learning of shared dependencies $\mathbf{A}$ from potential disturbances in structure personalization.

2.6 Balancing Personalization and Shared knowledge

In this sub-section, we first present how we make recommendation predictions based on the shared latent structure and the personalized latent structures, followed by illustrations of why and how we balance these two predictions for robust and effective recommendation.

Recommendation Prediction. The structured latent factors $\hat{\mathbf{e}}$ and $\hat{\mathbf{e}}^*$ obtained from shared structure learning and personalized structure learning encode both endogenous latent factors as well as their dependencies, which could help tell whether the user potentially prefers a particular target item. As such, we make the recommendation prediction based on $\hat{\mathbf{e}}$ and $\hat{\mathbf{e}}^*$ as follows:

$$\hat{Y}^c = MLP_R(\hat{\mathbf{e}}), \quad \hat{Y}^* = MLP_R^*(\hat{\mathbf{e}}^*), \tag{15}$$

where MLP_R and MLP_R^* are two prediction heads based on the shared latent structure and the personalized latent structure, respectively.

Balancing $\hat{Y}^c$ and $\hat{Y}^*$. Either $\hat{Y}^c$ or $\hat{Y}^*$ has its drawback in practice. Specifically, $\hat{Y}^c$ reflects an overall knowledge of all users and can hardly accommodate personalized or subjective preferences. Section 2.5 shows an intuitive example of user career and item price. $\hat{Y}^*$ alone might be less optimal for some users due to the well-known sparsity and noise problem of user behavior data in real-world recommender systems. For example, there might be less adequate data samples for optimal latent structure personalization

for some users. For these users, the personalization might become easily underfit or error-prone. Therefore, for each user, it is necessary to adaptively balance the prediction $\hat{Y}^*$ based on personalized latent structure and the prediction $\hat{Y}^c$ based on the shared latent structure.

To bridge the gap, we propose to explicitly estimate the uncertainty of the prediction $\hat{Y}^*$ based on personalized structures. For uncertainty estimation [27], an intuitive solution is to devise a bayesian neural network (BNN) with a probability distribution over the network parameters. Remember that the critical part of personalized structure learning is a probabilistic sampling (*cf*. Equation (12)) process based on the confidence for masking latent factors and shared dependencies. Therefore, we can estimate the expectation and variance of personalized prediction through multiple samples. Technically,

1) We obtain multiple masks $\{(\mathbf{M}^{(A),k}, \mathbf{M}^{(e),k})\}_{k=1}^{N_m}$ with different $g \sim \text{Uniform}(0, 1)$ via Equation (12).
2) Multiple personalized latent factors $\{\hat{\mathbf{e}}^{*,k}\}_{k=1}^{N_m}$ are generated through Equation (13)-(14) based on $\{(\mathbf{M}^{(A),k}, \mathbf{M}^{(e),k})\}_{k=1}^{N_m}$.
3) Multiple personalized predictions $\{\hat{Y}^{*,k}\}_{k=1}^{N_m}$ are obtained through Equation (15).

We estimate the expectation and the variance of personalized prediction as follows:

$$\hat{Y}_\mu^* = \frac{1}{N_m}\sum_k \hat{Y}^{*,k}, \quad \hat{Y}_\sigma^* = \frac{1}{N_m}\sum_k (\hat{Y}^{*,k} - \hat{Y}_\mu^*)^2. \quad (16)$$

The balanced prediction can be obtained by:

$$f_\Theta(u, i) = \hat{Y} = \hat{Y}^c * \sigma(\hat{Y}_\sigma^*/\beta) + \hat{Y}_\mu^* * (1 - \sigma(\hat{Y}_\sigma^*/\beta)), \quad (17)$$

where σ denotes the Sigmoid operation, and β is a normalization term that controls the smoothness of uncertainty distribution. $\sigma(\hat{Y}_\sigma^*/\beta)$ is the personalization uncertainty which indicates to what extent the personalized prediction is less trustworthy. The underlying intuition of this balancing is simple – when we are more uncertain about the current user (personalization), we trust more the decision-making patterns learned from all users (shared knowledge). $\hat{Y}$ is the final prediction score used for training models (*cf*. Equation (1)) and for ranking items at inference. In summary, the loss function used for training the entire model is:

$$\mathcal{L} = \mathcal{L}_R - ELBO + \mathcal{L}_{struct} + \mathcal{L}_{dag}, \quad (18)$$

where $\mathcal{L}_R$ is obtained with Equation (1).

Algorithm 1: Personalized Latent Structure Learning for Recommendation

Input: User-item interaction data $\mathcal{D} = \{(u, i, Y_{u,i})\}$ where $u \in \mathcal{U}$, $i \in \mathcal{I}$, and $Y_{u,i} \in \{0, 1\}$.
Initialize $\Theta = \{\phi, \theta, \mathbf{A}, \vartheta, MLP_{mask}, MLP_R, MLP_R^*\}$
repeat
  Sample a user-item interaction $(u, i, Y_{u,i}) \in \mathcal{D}$
  $\mathbf{z} \sim q_\phi(z \mid u, i)$
  $\hat{Y}^c = MLP_R(g_\vartheta(\mathbf{A}_i * ((I - \mathbf{A}^T)^{-1}\mathbf{z})))$
  for $k \leftarrow 0$ **to** N_m **do**
    Sample $\mathbf{M}^{(A),k}, \mathbf{M}^{(e),k}$ // Eq. (11) - (12)
    $\mathbf{e}^* = (I - (\mathbf{M}^{(A),k} * \mathbf{A}^T)^{-1})(\mathbf{M}^{(e),k} * \mathbf{z})$
    $\hat{Y}^{*,k} = MLP_R^*(g_\vartheta((\mathbf{M}_i^{(A),k} * \mathbf{A}_i) * (\mathbf{M}^{(e),k} * \mathbf{e}^*)))$
  end
  Calculate $\hat{Y}_\mu^*, \hat{Y}_\sigma^*, \hat{Y}$ // Eq. (16) - (17)
  $\Theta \leftarrow Update(\mathcal{L})$ // Eq. (18)
until *convergence*

3 Experiments

We conduct experiments on real-world datasets to answer three main research questions:

- **RQ1:** How does PlanRec perform compared to state-of-the-art VAE and disentanglement baselines in the recommendation domain, and state-of-the-art causal structure learning baselines in the generic domain?
- **RQ2:** How do different building blocks and different hyper-parameter settings affect PlanRec?
- **RQ3:** How does PlanRec internally benefit from uncertainty estimation and latent structure learning?

TABLE 1: Statistics of the Datasets.

Dataset	#Users	#Items	#Interactions	#Density
MovieLens	6, 040	3, 900	1, 000, 209	0.04246
Amazon	459, 133	313, 966	8, 898, 041	0.00063
Alipay	400, 594	19, 503	38, 710, 494	0.00495

3.1 Experimental Setup

Datasets. We conduct experiments on two real-world benchmark datasets, *i.e.*, MovieLens and Amazon Review, and a challenging industrial dataset Alipay, of which the statistics are summarized in Table 1.

- **Amazon.** We use the largest category, Amazon Book, and follow [22], [28] for evaluation. Specifically, interacted items are treated as positives, and randomly sampled items are negatives during training. We take the 16-core setting to ensure good dataset quality.
- **MovieLens.** We take the MovieLens-1M version and follow a similar setting as Amazon.
- **Alipay.** To validate model performance on large-scale industrial datasets, we collect the Alipay dataset from the Alipay platform[1] where applets, such as government services, are treated as items. Items exposed and clicked are treated as positives, while items exposed but not clicked are negatives during training.

Baselines. To have a comprehensive analysis, we consider the following three kinds of baselines for comparison.

- **Variational Auto-encoder**
  - *Mult-VAE* [9]. It uses VAE with multinomial likelihood on user-item implicit feedback data. We follow the official codebase[2] for implementation.
  - *ACVAE* [10]. It introduces adversarial learning and contrastive learning to improve the disentanglement of VAE on sequential recommendation. We follow the official codebase[3] for implementation.

[1]https://global.alipay.com/platform/site/ihome
[2]https://github.com/dawenl/vae_cf
[3]https://github.com/ACVAE/ACVAE-PyTorch

TABLE 2: A comparison between PlanRec and various kinds of baselines on three real-world datasets. We conduct two-sided test, and p-value < 0.05 indicates that the improvement over the best-performing baseline is statistically significant.

Dataset	Metric	Mult-VAE	DisenRec	ACVAE	DESTINE	VSAN	DAG-GNN	CVAE	PlanRec	Improv.	p-value
MovieLens	AUC	0.8334	0.8278	0.8236	0.8339	0.8376	0.8408	0.8443	**0.8545**	1.21%	4.81e-4
	Recall@5	0.3337	0.3301	0.3216	0.3369	0.3541	0.3635	0.3607	**0.3998**	9.99%	5.36e-5
	Recall@10	0.5027	0.4951	0.4801	0.5022	0.5175	0.5337	0.5360	**0.5625**	4.94%	2.07e-4
	NDCG@5	0.2568	0.2568	0.2492	0.2619	0.2756	0.2820	0.2796	**0.3136**	11.21%	7.07e-5
	NDCG@10	0.3369	0.3369	0.3260	0.3420	0.3549	0.3647	0.3648	**0.3928**	7.68%	1.97e-4
Amazon	AUC	0.8752	0.8748	0.8718	0.8769	0.8805	0.8803	0.8754	**0.9040**	2.67%	4.37e-5
	Recall@5	0.5485	0.5626	0.5046	0.5350	0.5504	0.5528	0.5621	**0.6292**	11.84%	7.39e-4
	Recall@10	0.6625	0.6726	0.6388	0.6529	0.6717	0.6805	0.6822	**0.7371**	8.05%	1.29e-5
	NDCG@5	0.4560	0.4837	0.4199	0.4518	0.4648	0.4645	0.4744	**0.5446**	12.59%	7.79e-4
	NDCG@10	0.5117	0.5373	0.4853	0.5093	0.5239	0.5268	0.5330	**0.5972**	11.15%	8.25e-4
Alipay	AUC	0.8495	0.8551	0.8588	0.8615	0.8680	0.8301	0.8280	**0.8754**	0.85%	2.23e-5
	Recall@5	0.3479	0.4142	0.3406	0.5001	0.4630	0.3687	0.3648	**0.5621**	12.40%	1.07e-4
	Recall@10	0.5672	0.5931	0.5837	0.6419	0.6328	0.5462	0.5590	**0.6822**	6.28%	6.31e-6
	NDCG@5	0.2437	0.2983	0.2452	0.4140	0.3564	0.2780	0.2627	**0.4743**	14.57%	1.80e-2
	NDCG@10	0.3508	0.3857	0.3642	0.4835	0.4396	0.3518	0.3578	**0.5330**	10.24%	3.62e-3

  - *VSAN* [8]. It encapsulates variational inference into self-attention based recommendation models. Since there is no public code release, we follow their paper details for implementation.
- **Disentanglement**
  - *DisenRec* [3]. This paper pretrains disentangled user representations through macro and micro disentanglement based on user behaviors. We follow the official codebase[4] for implementation.
  - *DESTINE* [13]. It decouples user-item feature learning into the pairwise term and the unary term. We follow the official codebase[5] for implementation.
- **Causal Structure Learning.** We choose two representative baselines in the generic domain and instantiate them on the Base model, *i.e.*, PlanRec without the proposed structure learning.
  - *DAG-GNN* [17]. It leverages graph neural networks [29] to recover the underlying directed acyclic graph (DAG) from observational data. We follow the official codebase[6] for implementation.
  - *CVAE* [30]. CVAE differs from [17] by using a neural structural causal model and requiring supervised signals to learn the disentanglement and causal structure. We follow the official codebase[7] for implementation.

Evaluation Protocal. All baselines and PlanRec involve complex interactions of the target item and user features, *i.e.*, focusing on the ranking phase of modern recommender systems. Therefore, we follow the evaluation protocol of ranking models [22], [23], [31]. Specifically, for each user, the last item in the behavior sequence is left for testing, and is considered as the positive item for training. One hundred randomly sampled items that the user did not consume are treated as negatives. We adopt three widely used metrics as the recommendation performance indicators, *i.e.*, Area Under the ROC Curve (AUC), Recall, and Normalized Discounted Cumulative Gain (NDCG). We consider the top 5/10 items ranked by models for metric computation. For all metrics, the larger values, the better.

Implementation Details. $q_\phi(z \mid u, i)$ is modeled as a two-layer self-attention network, which concatenates the behavior sequence and the target item as input. This network is followed by one fully-connected layer, which takes the last embedding of the output sequence of the self-attention network as input. Smoothing terms τ and β in Equation (12) and Equation (17) are set to 1 and 0.002, respectively. For all models, Adagrad [32] is used for optimization with learning rate 0.01 and batch size 4096. User/item embedding size is 8. For a fair comparison, we consider the embeddings of the last 50 clicked items as the user feature for all models.

3.2 Overall Performance (RQ1)

Table 2 shows the comparison results on three real-world datasets. We have the following observations:

- Mult-VAE is an early recommendation work that exploits variational inference to uncover latent factors for user-item data. The critical extension on vanilla VAE is the multinomial likelihood which they demonstrate is more suitable for implicit feedback data. It is a pioneer work but lacks advanced techniques for further disentangling users' interests, and capturing the dynamic user intention. As such, it mostly achieves inferior results compared to other baselines.
- DisenRec is built upon Mult-VAE by simultaneously considering the macro-disentanglement, which corresponds to the latent factor modeling in Mult-VAE, and the micro-disentanglement, which further forces different dimensions in a latent factor to encoder different fine-grained aspects. Therefore, DisenRec outperforms Mult-VAE on larger-scale datasets (Amazon and Alipay) where user interests are likely to be diverse and complex. Unexpectedly, ACVAE mostly achieves inferior results on three datasets. The proposed adversarial learning and contrastive learning are originally designed for user-item matching and might not be suitable for effectively modeling the complex interactions between a user and a target item in ranking.

[4]https://jianxinma.github.io/disentangle-recsys.html
[5]https://github.com/CRIPAC-DIG/DESTINE
[6]https://github.com/fishmoon1234/DAG-GNN
[7]https://github.com/huawei-noah/trustworthyAI

- DESTINE and VSAN are two baselines that leverage the advanced self-attention technique to capture users' dynamic and diverse interests. DESTINE achieves better performance by extending the self-attention module to consider unary prediction without feature interaction, which they claim is a disentanglement of the self-attention mechanism. VSAN effectively encapsulates transformer blocks to build the inference/generative module of VAE for user factor disentanglement. They outperform the above baselines by leveraging advanced user modeling and feature interaction techniques.
- DAG-GNN and CVAE are originally applied to protein signaling and computer vision, respectively. We take their structure learning modules to improve the dependency discovery of user-item latent factors while other components remain the same as ours. The apparent performance improvement over previous baselines in many cases demonstrates the necessity and effectiveness of latent structure learning for recommendation. However, we also observe that DAG-GNN and CVAE perform comparably worse on larger-scale datasets (*e.g.*, Amazon and AliPay) with significantly more users than on the MovieLens dataset. These results probably indicate that the universally learned causal structure might not be optimal for diverse users on larger datasets, demonstrating the need for personalized structure learning.
- PlanRec consistently outperforms various kinds of baselines across different datasets. Remarkably, PlanRec improves the best-performing baselines by 11.21%, 12.59%, and 14.57% *w.r.t.* NDCG@5 on MovieLens, Amazon, and Alipay, respectively. Interestingly, on larger scale datasets, such as Amazon and Alipay, where users' interests are likely to be more diverse and complex, structure learning related baselines (DAG-GNN, CVAE) cannot beat some disentanglement baselines, while PlanRec significantly outperforms all baselines. This result further demonstrates the necessity and effectiveness of the proposed personalized structure learning on real-world recommender systems. Besides, the proposed uncertainty estimation can effectively balance personalization and shared knowledge, accommodating the heterogeneity of diverse users and thus achieving consistent performance gains on both large-scale and small-scale datasets.. We conduct two-sided tests, and the improvements over the strongest baseline are all statistically significant with p-value less than 0.05.

3.3 Model Analysis (RQ2)

3.3.1 Analysis of N_m in Uncertainty Estimation.

We achieve personalization uncertainty estimation via probabilistically sampling N_m personalized graphs. We analyze how N_m affects the performance of PlanRec, and obtain the performance on Amazon and MovieLens with $N_M \in \{1, 2, 4, 8\}$. According to Table 3, we find that:

- $N_m = 1$ means a loss of uncertainty estimation, and the final prediction is obtained by the weighted sum of predictions from the shared structure and the personalized structure based on fixed weights. The large performance gap between $N_m = 1$ and $N_m = 2$ demonstrates the rationality of user-adaptive balancing of shared knowledge and personalization, and the effectiveness of our uncertainty estimation.
- Increasing N_m mostly leads to a performance improvement on MovieLens and Amazon. Different from MovieLens, on the larger-scale Amazon dataset, the performance change is less significant between $N_m = 4$ and $N_m = 8$, probably indicating that a small N_m is adequate for uncertainty estimation on larger-scale real-world datasets. This is one practical merit of PlanRec.

TABLE 3: Analysis of the number of sampled personalized structures N_m, as defined near Equation (16).

Dataset	Metric	$N_m = 1$	$N_m = 2$	$N_m = 4$	$N_m = 8$
Amazon	AUC	0.8960	0.8990	**0.9070**	0.9040
	R@5	0.6058	0.6226	**0.6317**	0.6292
	R@10	0.7177	0.7313	**0.7416**	0.7371
	N@5	0.5202	0.5385	**0.5468**	0.5446
	N@10	0.5748	0.5915	**0.6003**	0.5972
MovieLens	AUC	0.8443	0.8491	0.8542	**0.8545**
	R@5	0.3607	0.3849	0.3849	**0.3998**
	R@10	0.5360	0.5571	0.5562	**0.5625**
	N@5	0.2796	0.2992	0.3017	**0.3136**
	N@10	0.3648	0.3830	0.3833	**0.3928**

3.3.2 Robustness Analysis w.r.t. OOD Generalization.

We are interested in whether personalized latent structure learning could improve model robustness against distribution shift [33]. In this regard, we construct an out-of-distribution (OOD) version of the MovieLens dataset:

1) For each user, we split the behavior sequence in the middle, and obtain two sub-sequences. Each behavior associates an item with category annotations. We summarize the category distributions of two sub-sequences, and measure the mutual information of these two distributions. Intuitively, the mutual information score can serve as the ***interest consistency*** $w.r.t.$ item categories of a particular user, indicating to what extent the interests/intentions of this user change.
2) Intuitively, when the interests/intentions of a user change in testing, models encounter an out-of-distribution shift from training to testing. In this light, we select users with interest consistency scores less than the average score, and construct an OOD MovieLens dataset with their behaviors. We take the behaviors of the remaining users to construct a relatively IID dataset.
3) For each user in the OOD or IID MovieLens dataset, we take the first half of behaviors for training, and the second half of behaviors for testing.

To have a user-related shift, we follow [34] to construct a synthetic OOD recommendation dataset, where we simulate a purchasing power change for each user in testing. We also construct a synthetic IID dataset where the purchasing power of each user does not significantly change in testing. The testing results of the proposed PlanRec and the Base model (a VAE with latent factor disentanglement but without personalized structure learning) are listed in Table 4. We have the following observations:

TABLE 4: Analysis of the robustness against user interest shifts *w.r.t.* item category on the OOD MovieLens dataset and user purchasing power on the OOD synthetic dataset.

Dataset	Metric	IID		OOD	
		Base	PlanRec	Base	PlanRec
Synthetic	AUC	0.7992	0.8172	0.7794	0.8266
	R@5	0.2371	0.2630	0.1973	0.2483
	R@10	0.3809	0.4008	0.3336	0.4101
	N@5	0.1825	0.2199	0.1608	0.1970
	N@10	0.2521	0.2865	0.2260	0.2749
MovieLens	AUC	0.8469	0.8439	0.8178	0.8380
	R@5	0.3554	0.3757	0.2793	0.3290
	R@10	0.5296	0.5504	0.4448	0.5028
	N@5	0.2729	0.2900	0.2120	0.2506
	N@10	0.3575	0.3750	0.2923	0.3350

- Overall, PlanRec achieves consistent performance improvement over the Base model *w.r.t.* two causes of distribution shifts (*i.e.*, category interest shift, and purchasing power shift) in OOD settings and also IID settings. These results indicate the merits of PlanRec in different real-world recommendation scenarios.
- In OOD settings, two models achieve inferior results than in IID settings, indicating the challenges of confronting interest shifts in real-world scenarios. Compared to the performance gains in IID settings, PlanRec achieves significantly more gains in OOD settings over the Base model (*e.g.*, +.0497 in OOD and +.0203 in IID *w.r.t.* Recall@5 on the MovieLens dataset). These results further demonstrate the robustness of PlanRec *w.r.t.* OOD generalization. Through latent structure learning, PlanRec captures to what extent a particular user will rely on each decision-making pattern. For example, if PlanRec discovers that a user relies on $Career \rightarrow Price$, it will actively capture the purchasing power change inside the user's behavior sequence, and accordingly make recommendations.

3.3.3 Analysis of Latent Structure.

We are interested in whether the proposed disentanglement with latent structure learning is essential for recommendation. In this light, we conduct a quantitative analysis of the disentangled latent factors and the dependencies inside the shared latent structure.

Latent Factor Disentanglement Analysis. One of the major aims of latent structure learning is to extract effective latent factors from user behavior data and the target item embedding. PlanRec requires ***no*** supervision signals, such as physical attributes, for latent factor disentanglement. To evaluate whether PlanRec has achieved effective disentanglement, we force each latent factor to predict a particular physical attribute (such as user age, and item category) via classification/regression losses and evaluate whether physical features lead to *additional* improvement. Intuitively, less additional improvement probably indicates that the original disentanglement is more effective. We construct a **Base** VAE model without the proposed structure learning for comparison. We refer to PlanRec with user or item attribute prediction as **PlanRec w. u** or **PlanRec w. i**, respectively. According to the results listed in Table 5, we have the following findings:

TABLE 5: Latent factor disentanglement analysis.

Model	AUC	R@5	R@10	N@5	N@10
Base	0.8223	0.3238	0.4776	0.2514	0.3260
w. ui	0.8368	0.3450	0.5161	0.2666	0.3498
PlanRec	0.8545	0.3998	0.5625	0.3136	0.3928
w. u	0.8556	0.4088	0.5639	0.3205	0.3976
w. i	0.8555	0.4107	0.5646	0.3213	0.3964
w. ui	0.8580	0.4097	0.5691	0.3216	0.3993

TABLE 6: Disentangled dependency analysis.

Model	AUC	R@5	R@10	N@5	N@10
PlanRec	0.8545	0.3998	0.5625	0.3136	0.3928
w/o u2i	0.8490	0.3937	0.5448	0.3061	0.3790
w/o u2u & i2i	0.8532	0.4002	0.5490	0.3167	0.3867

- Both the Base model and PlanRec benefit from these external physical attributes, which help to make the latent factors meaningful and diversified, *i.e.*, better disentanglement. The improvement also shows that effective disentanglement is a contributing factor to recommendation performance.
- Compared to the performance gain of Base w. ui over Base, the performance gap between PlanRec and PlanRec w. ui is less notable. This finding probably indicates that the disentanglement in PlanRec is more effective by discovering latent factors that could approximate the utilities of physical user-item features.

Factor Dependency Disentanglement Analysis. We are interested in whether disentangling the factor dependencies is essential and how different kinds of dependencies (*e.g.*, $user \rightarrow item$ and $user \rightarrow user$) affect the performance of PlanRec. Therefore, we propose to explicitly mask two dependencies of elements in the structure with zero values, and obtain the results listed in Table 6. Specifically, we mask 1) dependencies between user factors and item factors, *i.e.*, **w/o u2i**; and 2) dependencies between user-user and item-item factors, *i.e.*, **w/o u2u & i2i**. It is noteworthy that when we mask more dependencies with zero values, we rely more on the disentangled latent factors themselves for prediction according to Equation (7).

We observe that removing any kind of dependencies leads to a performance drop, which means that explicitly modeling the dependencies of disentangled latent factors is helpful for recommendation, and that the proposed framework is effective. Removing the dependencies between user-item leads to a larger performance drop than removing dependencies between user-user or item-item latent factors. These results are intuitive since recommendation cares more about how and why users interact with items.

3.3.4 Ablation Studies.

To investigate how different building blocks affect PlanRec, we conduct ablation studies of critical architecture components and three regularizations, respectively.

Analysis of Architecture Components. We surgically add the following critical components onto the VAE base model

in a cumulative manner. The VAE base model makes predictions based on the disentangled latent factors using DNN. The results are shown in Table 8.

- **+ SSL** denotes adding the shared structure learning module and regularizations to the VAE base model. SSL explicitly models the dependencies between user-item latent factors (*e.g.*, $Career \rightarrow Price$), which indicate the decision-making patterns users in the recommender system typically rely on. The performance gain demonstrates the importance of the disentanglement of decision-making patterns for recommendation, and the effectiveness of our SSL module.
- **+ PSL** denotes further adding the personalized structure learning onto +SSL via probabilistic sampling. Note that we perform one-time sampling, *i.e.*, $N_m = 1$. Personalized structure learning enables the user-adaptive refinement on the shared structure learned from all users. The clear performance gain demonstrates the rationality of our analysis that some dependencies (*e.g.*, $Career \rightarrow Price$) might not be suitable for some users due to the unobserved private/sensitive factors (*e.g.*, family wealth), which can hardly be disentangled and captured in the shared structure. It also validates the effectiveness of our probabilistic design for structure personalization.
- **+ Uncer** means that we sample multiple personalized structures and accordingly estimate the uncertainty of predictions made on the personalized structures, as illustrated in Section 2.6. Uncertainty estimation of personalization leads to a balance between predictions made on shared and personalized structures. Not surprisingly, we observe a performance improvement by adding the uncertainty estimation.

Analysis of Structure Regularizations. We surgically add the three structure regularizations (*cf.* Section 2.4) onto a base model, *i.e.*, PlanRec without regularizations. The results are shown in Table 8.

- **+ RecR** denotes adding the reconstruction regularization, which ensures that the latent structure could imply the user decision-making patterns such that user interaction data could be reproduced. We observe consistent performance gains on two datasets, demonstrating the rationality of this regularization in latent structure learning.
- **+ AcyR** denotes further adding the acyclicity constraint, which ensures that there is no cycle in the latent structure; otherwise, some factors could both affect and be affected by some other factors. Not surprisingly, alleviating cyclic effects in latent structure learning leads to performance improvement.
- **+ DepR** denotes adding the dependency regularization, which ensures that ancestor latent factors could semantically cause the descendants. The clear performance gains reveal the necessity of enhanced dependency in the latent structure.

3.4 Case Studies (RQ3)

3.4.1 Analysis of Estimated Uncertainty.

To reveal the rationality of the uncertainty estimation, which eventually leads to the balancing of personalization and shared knowledge, we conduct experiments to uncover the correlation between users' activeness/interest-consistency and predicted uncertainty on personalization.

TABLE 7: Ablation study by adding critical components to the base model in a *cumulative* manner.

Dataset	Metric	VAE	+ SSL	+ PSL	+ Uncer
Amazon	AUC	0.8652	0.8740	0.8929	0.9040
	R@5	0.5322	0.5644	0.5948	0.6292
	R@10	0.6527	0.6829	0.7070	0.7371
	N@5	0.4483	0.4775	0.5101	0.5446
	N@10	0.5070	0.5353	0.5648	0.5972
MovieLens	AUC	0.8223	0.8423	0.8491	0.8545
	R@5	0.3238	0.3533	0.3849	0.3998
	R@10	0.4776	0.5305	0.5571	0.5625
	N@5	0.2514	0.2733	0.2992	0.3136
	N@10	0.3260	0.3594	0.3830	0.3928

TABLE 8: Ablation study by adding three structure regularizations to the base model in a *cumulative* manner.

Dataset	Metric	Base	+ RecR	+ AcyR	+ DepR
Amazon	AUC	0.8827	0.8931	0.8975	0.9040
	R@5	0.5938	0.6069	0.6135	0.6292
	R@10	0.6955	0.7096	0.7216	0.7371
	N@5	0.5141	0.5236	0.5301	0.5446
	N@10	0.5637	0.5761	0.5827	0.5972
MovieLens	AUC	0.8324	0.8443	0.8478	0.8545
	R@5	0.3437	0.3761	0.3820	0.3998
	R@10	0.4951	0.5297	0.5400	0.5625
	N@5	0.2723	0.2976	0.3066	0.3136
	N@10	0.3460	0.3721	0.3783	0.3928

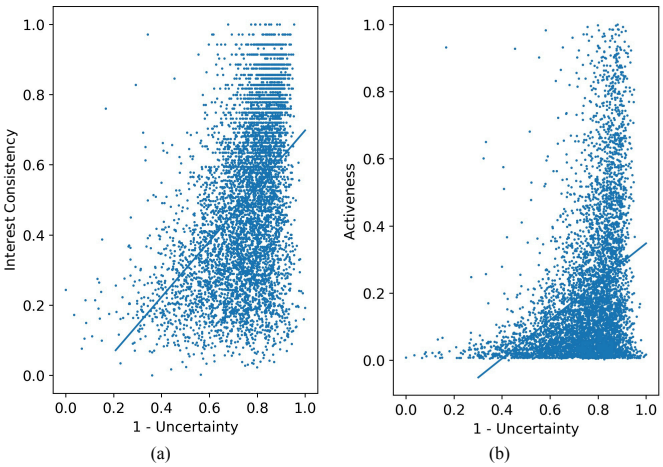

Fig. 3: A visualization of the correlation between users' interest consistency *w.r.t.* item categories (left) / activeness (right) and personalization uncertainty estimated by PlanRec. Lines are obtained using linear regression.

Interest Consistency and Uncertainty. For all the users on the MovieLens dataset, we take $\sigma(\hat{Y}^*_\sigma/\beta)$ estimated in Equation (16)-(17) as the ***uncertainty*** score. As for the computation of interest consistency *w.r.t.* item categories, we follow the procedure described in Section 3.3.2. For uncertainty and interest consistency scores, we linearly normalize the values by their maximum and minimum values. Then, we can obtain the interest consistency by (1 - uncertainty) scatter plot

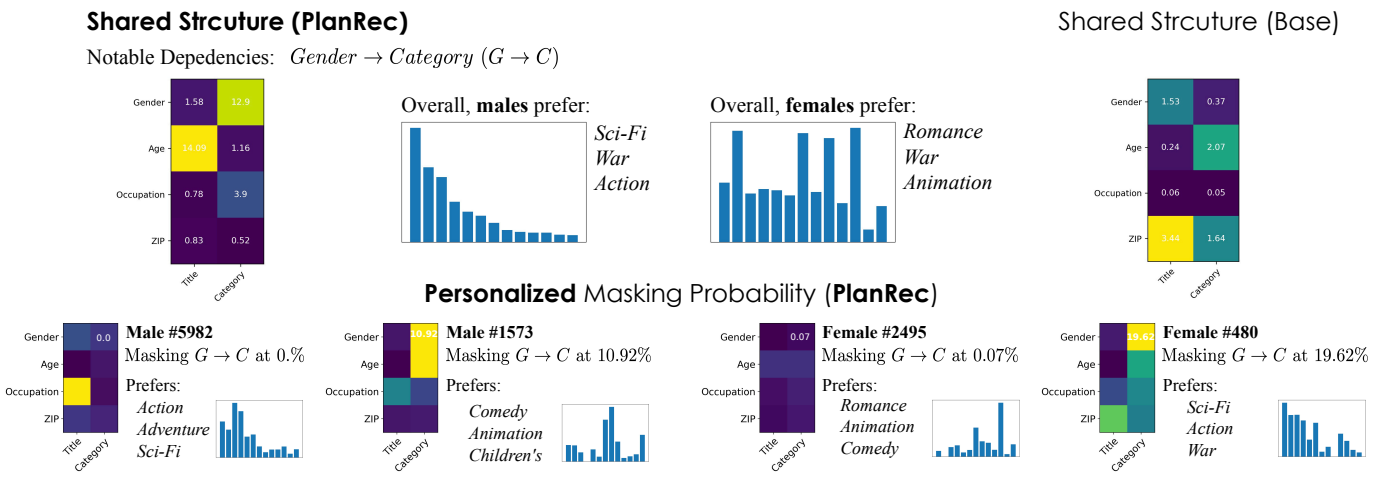

Fig. 4: A visualization of the learned shared structure and the personalized masking probabilities of four sampled users. We visualize the overall item category distributions of all male/female users, and the user-specific distributions.

as shown in Figure 3 (a). We observe that users with high interest consistency are mostly with low uncertainty, which is intuitive since such users are easier to model. PlanRec successfully and automatically captures this clue, and makes predictions mostly based on the personalized structure. The linear regression line reveals correlations between interest consistency and the predicted uncertainty, indicating the rationality of the uncertainty estimation in PlanRec.

Activeness and Uncertainty. Similarly, for all the users in the MovieLens dataset, we count the number of behaviors as the ***activeness***, and take $\sigma(\hat{Y}^*_\sigma/\beta)$ as the uncertainty. As such, we can obtain the activeness by (1 - uncertainty) scatter plot as shown in Figure 3 (b). For uncertainty and activeness scores, we linearly normalize the values with their maximum and minimum values. We observe that high activeness users are mostly with low uncertainty, which is intuitive since we can better capture a user's interests/intention when we have more behavior data.

As for low-activeness users, there seems to be no significant difference in their uncertainty. The reason probably is that many inactive users come to the platform looking for limited categories of items, *i.e.*, simple patterns that can be easily captured. Motivated by this supposition, we use a two-sided test to quantify the interest consistency difference of high-activeness (> 0.6) and low-activeness ($<= 0.6$) users. The t-statistic is 26.17 with p-value $9.45e-143$, showing the interest consistency of high-activeness users is statistically larger than that of low-activeness users. These results jointly demonstrate the effectiveness of our personalization and uncertainty estimation modules in PlanRec.

3.4.2 Analysis of Shared/Personalized Structures.

To explicitly analyze the learned shared/personalized structures, we visualize the dependencies of latent factors. Since latent factors lack human interpretability, we propose to force six latent factors to predict six physical user/item features of MovieLens via regression or classification. For example, we might add a linear layer that takes the first latent factor as input and learns to predict the user age via mean squared error. We also visualize the personalized masking probabilities *w.r.t.* the shared structure of four users. The results are shown in Figure 4. We observe that:

- From the visualized shared structure, we could notice some dependencies with high confidence, such as $Gender \rightarrow Category$ ($G \rightarrow C$). We further analyze the statistical correlation between $Gender$ and $Category$. We observe a statistically significant difference between male and female user behaviors in 10 out of 13 item categories using two-sided tests (p-value < 0.05). These results demonstrate that shared structure learning can discover potentially valid dependencies. In contrast, the Base model without latent structure learning fails to capture the $G \rightarrow C$ dependency from the user-item interaction data. We technically attribute this failure to the absence of the reconstruction, dependency, and acyclicity regularizations, without which the dependency matrix functions as other black-box parameters and lacks the desired properties and interpretability.
- We also visualize the personalized masking probability $\mathbf{M}^{(A)}$, as illustrated in Equation (12) before $0/1$ sampling, of four users. These users include two male users with different masking probabilities on the $G \rightarrow C$ dependency, *i.e.*, $0.\%$ and 10.92%, and two female users with similar differences. To have a better understanding of the results, we visualize the item category distribution[8] of these four users. We observe that users $\#5982, \#2495$ with similar distributions as the overall distribution of their corresponding gender are with **low** masking probabilities. On the contrary, users $\#1573, \#480$ with distributions significantly different from the overall distribution of their corresponding gender are with **high** masking probabilities.
- These results jointly demonstrate the rationality and effectiveness of personalized structure learning. Specifically, a high dependency score means that the model relies on users' $Gender$ and the target-items' $Category$ to measure whether users will interact with the tar-

[8]We normalize the category distributions by the prior distribution and remove five categories with the lowest number of items to alleviate dataset biases.

get items. As such, the model will further learn what item categories female users and male users prefer to interact with. The model will eventually rely on both the dependency $G \rightarrow C$ and the user preference of different genders to make predictions. However, for some users, such as Male $\#1573$ and Female $\#480$, models might draw false conclusions due to their personalized preferences. For these users, the personalized structure learning has a high probability of masking the dependency $G \rightarrow C$ and forces the model to rely on other dependencies, such as $Age \rightarrow Title$ and $Gender \rightarrow Title$ for predictions.

4 Related Works

Neural Recommender Systems. There is rapidly growing research interest in designing neural network based recommendation models [23], thanks to the breakthroughs in AI [35], [36], [37], [38]. Existing works leverage neural networks to perform representation learning of users and items, and model the complex interactions between them. Recent representation learning techniques include, but are not limited to, graph-based techniques [28], [39], [40] and sequence-based techniques [31], [41], [42]. To model the complex interactions of users and items or enhance representation learning, recent works incorporate advanced techniques ranging from Recurrent Neural Networks [41], [43], [44], attention mechanisms [31], [45], dynamic capsule routing [45], [46], to memory networks [47], [48].

Disentangled Representation Learning. Disentangled representation learning is an emerging technique that uncovers the underlying factors hidden in the observational data [12]. Early works have made efforts in the field of computer vision such as disentangling basic visual concepts with constrained variational inference [49]. Typical following works explore a better trade-off between disentanglement and reconstruction by encouraging factorial distribution [50], and providing the total correlation objective decomposed from the evidence lower bound [51]. Besides variational inference, other works exploit the attention mechanism [52] or multiple encoders such that one attention head or encoder could learn a mapping from observational data to one aspect [53].

Learning disentangled representations is a promising and nascent research direction of recommendation [1], [3], [4], [5], [6], [7], [8], [40], aiming to improve effectiveness and robustness. To ensure the variety of disentangled latent factors, Bayesian posterior inference and variational estimation [54] have been introduced into recommendation for latent factor disentanglement. [1] and [3] achieve macro-/micro- disentanglement to ensure factor-/dimension- level independence. Sharing similar motivations, [10] introduce adversarial learning and contrastive learning to reduce the correlation of latent factors, and different dimensions, respectively. Besides latent factor disentanglement, [13] disentangle the modeling of user/item features as the pair-wise terms and the unary terms. Additionally, some works explore improving disentangled recommendation by introducing auxiliary data, such as item-related vision-language [55] and category information [3]. We differ from existing works by primarily learning shared and personalized dependencies of latent factors. User behaviors are driven by latent factors and their interactions. Disentangling the latent structure can further improve recommendation robustness and effectiveness.

Causal Structure Learning. In the generic domain, causal structure learning, which uncovers causal relations and estimates the magnitude of these effects from relatively abundant observational data [15], [17], [18], [19], [19], [56], [57], [58], has recently gained much attention. Most of these approaches fall into the combinatoric group and are confronted with low-dimensional problems. More recently, there are a few continuous optimization methods dealing with high-dimensional problems, such as DEAR [59], CausalVAE [17], DCDI [60], meta-learning causal models [61] and a few related works that deal with non-semantic, high-dimensional data such as image, video data, and visual scenes [62], [63]. In spite of their significant advances, structure learning for discrete and relational user behavior data in recommender systems remains largely unexplored. In this paper, we are not theoretically doing causal structure learning, but devise causality-inspired techniques for recommendation, analogous to other domains [64], [65], [66], [67], [68], [69], [70], [71], [72]. We resort to some key ideas in causal structure learning for improving the dependency discovery of latent structure learning for recommendation. Moreover, we summarize the key challenges of applying shared latent structure for recommendation, and devise the personalized structure learning framework as well as the balancing of personalization and shared knowledge.

5 Conclusion and Future Work

In this paper, we study how to jointly uncover user-item latent factors and their dependencies, *i.e.*, latent structure learning. We summarize the key challenges of latent structure learning in recommendation, and demonstrate the necessity of structure personalization and the balancing of personalization and shared knowledge. We propose the personalized latent structure learning framework for recommendation, namely PlanRec. PlanRec involves structure personalization via probabilistic sampling conditioned on user embedding, and uncertainty estimation inspired by bayesian neural networks. We conduct empirical studies on three real-world datasets, and provide insightful analyses on the rationality of PlanRec.

This work uncovers the interactions of users and items latent factors behind the decision-making patterns of all users via shared structure learning, and behind a particular user via personalized structure learning. We believe this new paradigm can be inspirational to the solving of some long-standing recommendation problems. For example, we will explore whether the uncovered latent factors and their dependencies can help alleviate bias issues in recommender systems, such as disentangling popularity bias and dealing with the dependencies between the popularity factor and the others. Moreover, we are interested in extending PlanRec to content-based recommendation models, where disentangling the user and content latent factors as well as their dependencies can better uncover users' decision-making

patterns, pursuing more effective and robust models in complex real-world scenarios.

ACKNOWLEDGMENTS

The work is supported by the National Natural Science Foundation of China (No. U20A20387, 62037001), National Key R & D Projects of the Ministry of Science and Technology (2020YFC0832500), Zhejiang Natural Science Foundation (No. LR19F020006), and Project by Shanghai AI Laboratory (No. P22KS00111).

REFERENCES

[1] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," in *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, 2019, pp. 5712–5723.

[2] L. Hu, S. Xu, C. Li, C. Yang, C. Shi, N. Duan, X. Xie, and M. Zhou, "Graph neural news recommendation with unsupervised preference disentanglement." in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[3] X. Wang, H. Chen, Y. Zhou, J. Ma, and W. Zhu, "Disentangled representation learning for recommendation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] Y. Wang, S. Tang, Y. Lei, W. Song, S. Wang, and M. Zhang, "Disenhan: Disentangled heterogeneous graph attention network for recommendation," in *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*. ACM, 2020, pp. 1605–1614.

[5] P. Nema, A. Karatzoglou, and F. Radlinski, "Disentangling preference representations for recommendation critiquing with ß-vae." in *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, 2021.

[6] X. Wang, H. Chen, and W. Zhu, "Multimodal disentangled representation for recommendation," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2021, pp. 1–6.

[7] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, "Disentangling user interest and conformity for recommendation with causal embedding." in *WWW '21: The Web Conference 2021*, 2021.

[8] J. Zhao, P. Zhao, L. Zhao, Y. Liu, V. S. Sheng, and X. Zhou, "Variational self-attention network for sequential recommendation." in *37th IEEE International Conference on Data Engineering, ICDE 2021*, 2021.

[9] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering." in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018*, 2018.

[10] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, and Y. Ding, "Adversarial and contrastive variational autoencoder for sequential recommendation." in *WWW '21: The Web Conference 2021*, 2021.

[11] A. Achille and S. Soatto, "Information dropout: Learning optimal representations through noisy computation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2897–2905, 2018.

[12] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[13] Y. Xu, Y. Zhu, F. Yu, Q. Liu, and S. Wu, "Disentangled self-attentive neural networks for click-through rate prediction." in *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management*, 2021.

[14] Y. Zhang, Z. Zhu, Y. He, and J. Caverlee, "Content-collaborative disentanglement representation learning for enhanced recommendation," in *Fourteenth ACM Conference on Recommender Systems*, 2020, pp. 43–52.

[15] P. O. Hoyer, S. Shimizu, A. J. Kerminen, and M. Palviainen, "Estimation of causal effects using linear non-gaussian causal models with hidden variables," *International Journal of Approximate Reasoning*, 2008.

[16] H. Ma, K. Aihara, and L. Chen, "Detecting causality from nonlinear dynamics with short-term time series," *Scientific reports*, vol. 4, no. 1, pp. 1–10, 2014.

[17] Y. Yu, J. Chen, T. Gao, and M. Yu, "Dag-gnn: Dag structure learning with graph neural networks." in *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, 2019.

[18] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, "Dags with no tears: Continuous optimization for structure learning," *arXiv preprint arXiv:1803.01422*, 2018.

[19] T. L. Griffiths and J. B. Tenenbaum, "Theory-based causal induction." *Psychological review*, 2009.

[20] B. Huang, K. Zhang, P. Xie, M. Gong, E. P. Xing, and C. Glymour, "Specific and shared causal relation modeling and mechanism-based clustering," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[21] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[22] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction." in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*, 2018.

[23] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering." in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*, 2017.

[24] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. J. Kerminen, "A linear non-gaussian acyclic model for causal discovery," *J. Mach. Learn. Res.*, vol. 7, pp. 2003–2030, 2006.

[25] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax." in *5th International Conference on Learning Representations, ICLR 2017*, 2017.

[26] Y. Bengio, N. Léonard, and A. C. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *CoRR*, vol. abs/1308.3432, 2013.

[27] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. M. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *CoRR*, vol. abs/2107.03342, 2021.

[28] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering." in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019*, 2019.

[29] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs." in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[30] M. Yang, F. Liu, Z. Chen, X. Shen, J. Hao, and J. Wang, "Causalvae: Disentangled representation learning via neural structural causal models." in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, 2021.

[31] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer." in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*, 2019.

[32] J. C. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *J. Mach. Learn. Res.*, 2011.

[33] R. Christiansen, N. Pfister, M. E. Jakobsen, N. Gnecco, and J. Peters, "A causal framework for distribution generalization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[34] W. Wang, X. Lin, F. Feng, X. He, M. Lin, and T. Chua, "Causal representation learning for out-of-distribution recommendation," in *WWW '22: The ACM Web Conference 2022*, pp. 3562–3571.

[35] Y. Zhuang, M. Cai, X. Li, X. Luo, Q. Yang, and F. Wu, "The next breakthroughs of artificial intelligence: The interdisciplinary nature of ai," *Engineering*, vol. 6, no. 3, p. 245, 2020.

[36] Y. Pan, "Multiple knowledge representation of artificial intelligence," *Engineering*, vol. 6, no. 3, pp. 216–217, 2020.

[37] Y.-t. Zhuang, F. Wu, C. Chen, and Y.-h. Pan, "Challenges and opportunities: from big data to knowledge in ai 2.0," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, no. 1, pp. 3–14, 2017.

[38] Y.-G. Lyu, "Artificial intelligence: Enabling technology to empower society," *Engineering*, vol. 6, no. 3, pp. 205–206, 2020.

[39] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020*, 2020.

[40] X. Wang, H. Jin, A. Zhang, X. He, T. Xu, and T.-S. Chua, "Disentangled graph collaborative filtering." in *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, 2020.

[41] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks." in *4th International Conference on Learning Representations, ICLR 2016*, 2016.

[42] S. Zhang, D. Yao, Z. Zhao, T.-S. Chua, and F. Wu, "Causerec: Counterfactual user sequence synthesis for sequential recommendation." in *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[43] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations." in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018*, 2018.

[44] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks." in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017*, 2017.

[45] Y. Cen, J. Zhang, X. Zou, C. Zhou, H. Yang, and J. Tang, "Controllable multi-interest framework for recommendation." in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2020.

[46] C. Li, Z. Liu, M. Wu, Y. Xu, H. Zhao, P. Huang, G. Kang, Q. Chen, W. Li, and D. L. Lee, "Multi-interest network with dynamic routing for recommendation at tmall." in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*, 2019.

[47] X. Chen, H. Xu, Y. Zhang, J. Tang, Y. Cao, Z. Qin, and H. Zha, "Sequential recommendation with user memory networks." in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018*, 2018.

[48] J. Huang, W. X. Zhao, H. Dou, J.-R. Wen, and E. Y. Chang, "Improving sequential recommendation with knowledge-enhanced memory networks." in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018*, 2018.

[49] I. Higgins, L. Matthey, A. Pal, C. P. Burgess, X. Glorot, M. M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[50] H. Kim and A. Mnih, "Disentangling by factorising," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, vol. 80, pp. 2654–2663.

[51] T. Q. Chen, X. Li, R. B. Grosse, and D. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 2018, pp. 2615–2625.

[52] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An unsupervised neural attention model for aspect extraction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. Association for Computational Linguistics, 2017, pp. 388–397.

[53] S. Jain, E. Banner, J. van de Meent, I. J. Marshall, and B. C. Wallace, "Learning disentangled representations of texts with application to biomedical abstracts," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2018, pp. 4683–4693.

[54] D. P. Kingma and M. Welling, "Auto-encoding variational bayes." in *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[55] X. Wang, H. Chen, and W. Zhu, "Multimodal disentangled representation for recommendation," in *2021 IEEE International Conference on Multimedia and Expo, ICME 2021, Shenzhen, China, July 5-9, 2021*. IEEE, 2021, pp. 1–6.

[56] S. Zhu, I. Ng, and Z. Chen, "Causal discovery with reinforcement learning." in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[57] S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien, "Gradient-based neural dag learning." in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[58] T. Kyono, Y. Zhang, and M. van der Schaar, "CASTLE: regularization via auxiliary causal graph discovery," in *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, 2020.

[59] X. Shen, F. Liu, H. Dong, Q. Lian, Z. Chen, and T. Zhang, "Disentangled generative causal representation learning," *CoRR*, vol. abs/2010.02637, 2020.

[60] P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien, and A. Drouin, "Differentiable causal discovery from interventional data," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[61] N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, C. Pal, and Y. Bengio, "Learning neural causal models from unknown interventions," *CoRR*, vol. abs/1910.01075, 2019.

[62] D. Bear, C. Fan, D. Mrowca, Y. Li, S. Alter, A. Nayebi, J. Schwartz, L. Fei-Fei, J. Wu, J. Tenenbaum, and D. L. K. Yamins, "Learning physical graph representations from visual scenes," in *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, 2020.

[63] S. Ehrhardt, O. Groth, A. Monszpart, M. Engelcke, I. Posner, N. J. Mitra, and A. Vedaldi, "RELATE: physically plausible multi-object scene synthesis using structured latent spaces," in *Advances in Neural Information Processing Systems 33, NeurIPS 2020*, 2020.

[64] X. Yang, H. Zhang, and J. Cai, "Deconfounded image captioning: A causal retrospect," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[65] W. Shi, G. Huang, S. Song, and C. Wu, "Temporal-spatial causal interpretations for vision-based reinforcement learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[66] H. Zhang, L. Xiao, X. Cao, and H. Foroosh, "Multiple adverse weather conditions adaptation for object detection via causal intervention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[67] X. Wang, Y. Wu, A. Zhang, F. Feng, X. He, and T.-S. Chua, "Reinforced causal explainer for graph neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[68] W. Wang, J. Gao, and C. Xu, "Weakly-supervised video object grounding via causal intervention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[69] S. Zhang, X. Feng, W. Fan, W. Fang, F. Feng, W. Ji, S. Li, W. Li, S. Zhao, Z. Zhao, T.-S. Chua, and F. Wu, "Video audio domain generalization via confounder disentanglement," in *AAAI*, 2023.

[70] S. Zhang, T. Jiang, T. Wang, K. Kuang, Z. Zhao, J. Zhu, J. Yu, H. Yang, and F. Wu, "Devlbert: Learning deconfounded visio-linguistic representations," in *MM '20: The 28th ACM International Conference on Multimedia*. ACM, 2020, pp. 4373–4382.

[71] X. Qian, Y. Xu, F. Lv, S. Zhang, Z. Jiang, Q. Liu, X. Zeng, T. Chua, and F. Wu, "Intelligent request strategy design in recommender system," in *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2022, pp. 3772–3782.

[72] K. Kuang, L. Li, Z. Geng, L. Xu, K. Zhang, B. Liao, H. Huang, P. Ding, W. Miao, and Z. Jiang, "Causal inference," *Engineering*, vol. 6, no. 3, pp. 253–263, 2020.

Shengyu Zhang is currently a Ph.D. student at Zhejiang University, Hangzhou, China. He got the China Scholarship Council Fellowship in 2020 and visited National University of Singapore, Singapore, as a visiting research scholar from 2021 to 2022. His current research interests are in information retrieval, and causally reguarlized machine learning. Until now, he has published more than 20 papers in major international journals and conferences such as SIGKDD, SIGIR, ACM MM, WWW, CVPR, AAAI, and TKDE, etc. He also serves as Reviewer of several high-level international journals such as TKDE, TCYB, TNNLS, etc.

Fuli Feng is a professor at the University of Science and Technology of China (USTC). He received Ph.D. in Computer Science from NUS in 2019. His research interests include information retrieval, data mining, and multi-media processing. He has over 30 publications appeared in several top conferences such as SIGIR, WWW, and MM, and journals including TKDE and TOIS. His work on Bayesian Personalized Ranking has received the Best Poster Award of WWW 2018. Moreover, he has been served as the PC member for several top conferences including SIGIR, WWW, WSDM, NeurIPS, AAAI, ACL, MM, and invited reviewer for prestigious journals such as TOIS, TKDE, TNNLS, TPAMI, and TMM.

Kun Kuang received his Ph.D. degree from Tsinghua University in 2019. He is now an Associate Professor in the College of Computer Science and Technology, Zhejiang University. He was a visiting scholar with Prof. Susan Athey's Group at Stanford University. His main research interests include Causal Inference, Artificial Intelligence, and Causally Regularized Machine Learning. He has published over 40 papers in major international journals and conferences, including SIGKDD, ICML, ACM MM, AAAI, IJCAI, TKDE, TKDD, Engineering, and ICDM, etc.

Wenqiao Zhang received the Ph.D. degree from the Zhejiang University, Hangzhou, China, in 2021. He is a Research Fellow with the College of Computer Science, National University of Singapore, Singapore. His current research interests include cross-media analysis and computer-aided healthcare. So far, he has authored many papers in the top-tier scientific journal/conference such as the TMM, TVCG, AAAI, CVPR, ACL, ACM-MM, KDD.

Zhou Zhao (Member, IEEE) received the B.S. and Ph.D. degrees in computer science from The Hong Kong University of Science and Technology, in 2010 and 2015, respectively. He is currently an Associate Professor with the College of Computer Science, Zhejiang University. His research interests include machine learning and data mining.

Hongxia Yang received the PhD degree in statistics from Duke University, in 2010. She is working as a senior staff data scientist and director with the Alibaba Group. She has published more than 60 papers and held nine filed/to be filed US patents and is serving as the associate editor of Applied Stochastic Models in Business and Industry.

Tat-Seng Chua is the KITHCT Chair Professor at the School of Computing, National University of Singapore. He was the Acting and Founding Dean of the School during 1998-2000. Dr Chua's main research interest is in multimedia information retrieval and social media analytics. In particular, his research focuses on the extraction, retrieval and question-answering (QA) of text and rich media arising from the Web and multiple social networks. He is the co-Director of NExT, a joint Center between NUS and Tsinghua University to develop technologies for live social media search. Dr Chua is the 2015 winner of the prestigious ACM SIGMM award for Outstanding Technical Contributions to Multimedia Computing, Communications and Applications. He is the Chair of steering committee of ACM International Conference on Multimedia Retrieval (ICMR) and Multimedia Modeling (MMM) conference series. Dr Chua is also the General Co-Chair of ACM Multimedia 2005, ACM CIVR (now ACM ICMR) 2005, ACM SIGIR 2008, and ACM Web Science 2015. He serves in the editorial boards of four international journals. Dr. Chua is the co-Founder of two technology startup companies in Singapore. He holds a PhD from the University of Leeds, UK.

Fei Wu (Senior Member, IEEE) received the Ph.D. degree from Zhejiang University, Hangzhou, China. He was a Visiting Scholar with the Prof. B. Yu's Group, University of California at Berkeley, Berkeley, from 2009 to 2010. He is currently a Full Professor with the College of Computer Science and Technology, Zhejiang University. His current research interests include multimedia retrieval, sparse representation, and machine learning.