Knowledge-Guided Disentangled Representation Learning for Recommender Systems

SHANLEI MU, Renmin University of China
YALIANG LI, Alibaba Group
WAYNE XIN ZHAO, SIQING LI, and JI-RONG WEN, Renmin University of China

In recommender systems, it is essential to understand the underlying factors that affect user-item interaction. Recently, several studies have utilized disentangled representation learning to discover such hidden factors from user-item interaction data, which shows promising results. However, without any external guidance signal, the learned disentangled representations lack clear meanings, and are easy to suffer from the data sparsity issue.

In light of these challenges, we study how to leverage knowledge graph (KG) to guide the disentangled representation learning in recommender systems. The purpose for incorporating KG is twofold, making the disentangled representations interpretable and resolving data sparsity issue. However, it is not straightforward to incorporate KG for improving disentangled representations, because KG has very different data characteristics compared with user-item interactions. We propose a novel **K**nowledge-guided **D**isentangled **R**epresentations approach (*KDR*) to utilizing KG to guide the disentangled representation learning in recommender systems. The basic idea, is to first learn more interpretable disentangled dimensions (explicit disentangled representations) based on structural KG, and then align implicit disentangled representations learned from user-item interaction with the explicit disentangled representations. We design a novel alignment strategy based on mutual information maximization. It enables the KG information to guide the implicit disentangled representation learning, and such learned disentangled representations will correspond to semantic information derived from KG. Finally, the fused disentangled representations are optimized to improve the recommendation performance. Extensive experiments on three real-world datasets demonstrate the effectiveness of the proposed model in terms of both performance and interpretability.

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

Additional Key Words and Phrases: Knowledge graph, recommender system, representation learning, disentangled representation

6

This work was partially supported by the National Natural Science Foundation of China under Grant No. 61872369 and 61832017, Beijing Academy of Artificial Intelligence (BAAI) under Grant No. BAAI2020ZJ0301, Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098, the Fundamental Research Funds for the Central Universities, and the Research Funds of Renmin University of China under Grant No.18XNLG22 and 19XNQ047. This work was also supported by Alibaba Group through Alibaba Innovative Research Program and Alibaba Research Intern Program.
Authors' addresses: S. Mu, School of Information, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China; email: slmu@ruc.edu.cn; Y. Li, Alibaba Group, Bellevue, WA, USA; email: yaliang.li@alibaba-inc.com; W. X. Zhao (corresponding author) and J.-R. Wen, Gaoling School of Artificial Intelligence, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China; emails: {batmanfly, jirong.wen}@gmail.com; S. Li, Renmin University of China, Beijing, 100872, China; email: lisiqing@ruc.edu.cn.

1046-8188/2021/08-ART6 $15.00
https://doi.org/10.1145/3464304

ACM Reference format:
Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Siqing Li, and Ji-Rong Wen. 2021. Knowledge-Guided Disentangled Representation Learning for Recommender Systems. *ACM Trans. Inf. Syst.* 40, 1, Article 6 (August 2021), 26 pages.
https://doi.org/10.1145/3464304

1 INTRODUCTION

Nowadays, recommender systems play an indispensable role in meeting users' personalized interests and alleviating the information overload problem. For developing effective recommendation approaches, representation learning has become an important technique that aims to learn good representations of both users and items. Most of the existing studies [16, 17, 23, 34, 57] represent users and items in the form of a single vector learned from the user-item interactions. However, it is widely recognized that the user behavior for item, adoption is a complex decision process [9, 32, 53], which is potentially affected by many potential factors.

Single-vector representations are infeasible to directly capture fine-grained user preferences over different factors, which limits the performance and interpretability of recommender systems. Intuitively, it is promising to learn better representations of users and items if we can disentangle and capture different latent factors underlying the user-item interactions. To fulfill this goal, several recent efforts have been devoted to learning disentangled representations for users and items in recommender systems [32, 53]. The disentangled representation learning based on the user-item interaction data can produce multiple-aspect[1] representations to model user preference and item, characteristics in different aspects (i.e., factors). However, the disentangled representations learned by existing methods lack clear meanings or interpretations. For example, it is usually difficult to infer the corresponding aspect that some specific disentangled vector is associated with. Meanwhile, as it learns multiple-aspect representations, disentangled representation learning is more likely to suffer from the data sparsity issue, especially for inactive users or unpopular items. Therefore, it poses two challenges to learn effective disentangled representations in recommender systems: making disentangled representations more interpretable and resolving data sparsity issue.

To tackle these challenges, in this article, we propose to utilize **knowledge graph** (**KG**) data to improve the disentangled representation learning in recommender systems. As we know, there is a large number of structured facts about entities in KG, and thus we can leverage rich entity information to tackle the aforementioned challenges in the disentangled representation learning. First, KG stores entity information in the form of multiple aspects or relations, which can be considered as explainable factors for the user-item interactions. For example, in the scenario of movie recommender systems, a user selects a movie to watch as the director of this movie is her/his favorite (i.e., due to the factor of *director*), or she/he just likes science fiction movies (i.e., due to the factor of *genre*). With the guidance of KG, it is feasible to learn disentangled representations with interpretable meanings. Second, the KG information can be used as auxiliary information to alleviate the data sparsity issue in recommender systems. Intuitively, KG can be used to enrich the context data and derive additional supervision signals for recommendation algorithms.

Indeed, KG data has been widely explored in recommender systems. Early methods [1, 22, 58, 60] apply **KG embedding** (**KGE**) techniques [50] to encode the information of KG into low-dimensional vectors for enriching the representations of users or items. Later on, another group of methods [21, 40, 54, 56] utilizes the structure of KG to construct some connectivity patterns as features to enrich the representations of users and items. Recently, several methods [46, 51]

[1]The term *aspect* refers to an attribute dimension of an entity, e.g., price, service, and location for restaurants.

construct a collaborative KG using both user-item interactions and KG, and then apply **graph neural network** (**GNN**) to learn the representations of users and items. While, these existing studies mainly focus on learning a unified single-vector representation that encodes useful KG information for recommendation. In other words, they aim to fuse the information from different aspects rather than disentangle the coupled information. To the best of our knowledge, there are seldom studies in which KG data is utilized to improve the disentangled representation learning in recommender systems.

To this end, we propose a novel approach to utilizing KG to guide the disentangled representation learning in recommender systems, named as **Knowledge-guided Disentangled Representations** (KDR). By leveraging KG data, we design an effective fusion way that can enhance the capacity and interpretability of the disentangled representation learning. Our approach consists of three modules: implicit disentangled representation module, explicit disentangled representation module, and representation alignment module. In the first two modules, KDR learns *implicit disentangled representations* and *explicit disentangled representations* from user-item interaction data and KG data, respectively. Then in the representation alignment module, we adopt the mutual information maximization strategy to align explicit and implicit disentangled aspects, so that implicit disentangled representation learning can be guided via structural KG semantics. By combining these three modules, our approach can learn more interpretable and effective disentangled representations *w.r.t.* multiple aspects for recommender systems.

To evaluate our KDR approach, we conduct extensive experiments on three real-world datasets from different application domains. Experimental results demonstrate that the proposed KDR approach achieves better performance compared to several competitive recommendation methods, especially in the sparse setting. We also conduct ablation studies to examine the usefulness of KG in the process of learning disentangled representations. Furthermore, case studies from real-world applications also demonstrate that our approach is more interpretable with aligned disentangled aspects.

To summarize, the main contributions of this article are as follows:

- To the best of our knowledge, it is the first time that structural KG data has been utilized to improve disentangled representation learning in recommender systems. Such an idea, is important to improve both effectiveness and interpretability of recommendation algorithms.
- We propose a novel approach to implementing the knowledge-guided disentangled representation learning. Specially, we learn implicit disentangled representations and explicit disentangled representations from user-item interaction data and KG data, respectively. And, we propose a mutual information maximization strategy to align explicit and implicit disentangled aspects.
- We conduct extensive experiments on several real-world datasets to validate the effectiveness of the proposed method, and demonstrate its advantages in terms of performance and interpretability.

The rest of this article is organized as follows. In Section 2, we introduce the background for this work and define our task. Section 3 presents the proposed knowledge-guided disentangled representations. The experimental setup and results are presented in Section 4. In Section 5, we review the related works in two aspects. Finally, Section 6 concludes the article.

2 PROBLEM FORMULATION

In this section, we first introduce the used notations throughout the article, and then define the studied task. We summarized the used notations in Table 1.

Table 1. Notations and Explanations

Notation	Explanation
u	user
i	item,
$\mathcal{U}$	the set of users
$\mathcal{I}$	the set of items
$\mathcal{G}_{rs}$	the user-item interaction graph
$\mathcal{G}_{kg}$	the knowledge graph
$\mathcal{E}_I$	the item, entity set
$\mathcal{E}_A$	the attribute entity set
$\mathcal{R}$	the relation set
$\boldsymbol{v}_{u,k} \in \mathbb{R}^d$	the final disentangled representation of user u at aspect k
$\boldsymbol{v}_{i,k} \in \mathbb{R}^d$	the final disentangled representation of item, i at aspect k
$\boldsymbol{h}_{u,k} \in \mathbb{R}^d$	the implicit disentangled representation of user u at aspect k
$\boldsymbol{h}_{i,k} \in \mathbb{R}^d$	the implicit disentangled representation of item, i at aspect k
$\boldsymbol{z}_{u,k} \in \mathbb{R}^d$	the explicit disentangled representation of user u at aspect k
$\boldsymbol{z}_{i,k} \in \mathbb{R}^d$	the explicit disentangled representation of item, i at aspect k
$\boldsymbol{x} \cdot \boldsymbol{y}$	the inner product between vector $\boldsymbol{x}$ and vector $\boldsymbol{y}$
$\hat{y}_{u,i}$	the matching score between user u and item, i

User-Item Interaction Graph. In recommender systems, we usually have historical user-item interaction information (e.g., views, clicks, and purchases). Here, we represent user-item interaction data in recommender system as a bipartite graph $\mathcal{G}_{rs} = \{(u,i)|u \in \mathcal{U}, i \in \mathcal{I}\}$, where $\mathcal{U}$ and $\mathcal{I}$ denote the set of users and the set of items, respectively. An edge, $(u,i) \in \mathcal{G}_{rs}$ indicates that there is an observed interaction record between user u and item, i.

Knowledge Graph. A KG contains rich attribute or relation information for entities in the form of *triples*, denoted by $\mathcal{G}_{kg} = \{\langle h,r,t\rangle | h,t \in \mathcal{E}, r \in \mathcal{R}\}$, where $\mathcal{E}$ and $\mathcal{R}$ denote the entity set and relation set. A triple $\langle h,r,t\rangle$ describes that there is a relation r between head entity h and tail entity t. Following [22, 51], we establish a set of item-entity linkages between KG and user-item interaction graph. The entities that can be linked to items in recommender systems are called *item entities*, and other entities describing *item entities*' attribute information in triples are called *attribute entities*. Thus, the entity set $\mathcal{E}$ can be divided into two sets, namely *item entity set* $\mathcal{E}_I$ and *attribute entity set* $\mathcal{E}_A$. For example, given a triple ⟨Avatar, directedBy, JamesCameron⟩ for movie recommender systems, *Avatar* is an item, entity, *James Cameron* is an attribute entity and *directedBy* denotes the relation between the two entities. Note that the rest entities (e.g., high-order connected entities to items) can be either removed or kept in KG. For clarity, we only discuss item, entities and attribute entities in our formulation, while the rest entities (e.g., high-order connected entities to items) can be easily incorporated.

Multi-Aspect Representation. Following disentangled representation learning [31], we represent entities with multi-aspect embeddings. Formally, we assume that there are total K aspects, each of which corresponds to one or some factor(s). Then, we embed user u, item, i, and attribute entity a into $\boldsymbol{v}_u = [\boldsymbol{v}_{u,1}, \boldsymbol{v}_{u,2}, \ldots, \boldsymbol{v}_{u,K}] \in \mathbb{R}^{K\times d}$, $\boldsymbol{v}_i = [\boldsymbol{v}_{i,1}, \boldsymbol{v}_{i,2}, \ldots, \boldsymbol{v}_{i,K}] \in \mathbb{R}^{K\times d}$, and $\boldsymbol{v}_a = [\boldsymbol{v}_{a,1}, \boldsymbol{v}_{a,2}, \ldots, \boldsymbol{v}_{a,K}] \in \mathbb{R}^{K\times d}$, respectively. Here, $\boldsymbol{v}_{u,k}$, $\boldsymbol{v}_{i,k}$ and $\boldsymbol{v}_{a,k}$ are d-dimensional vectors, denoting user, item, and attribute embedding for the k th aspect.

Task Definition. Given a user u, the studied task aims to recommend top-N items from the candidate items based on both the user-item interaction graph $\mathcal{G}_{rs}$ and the KG $\mathcal{G}_{kg}$. Specifically, a

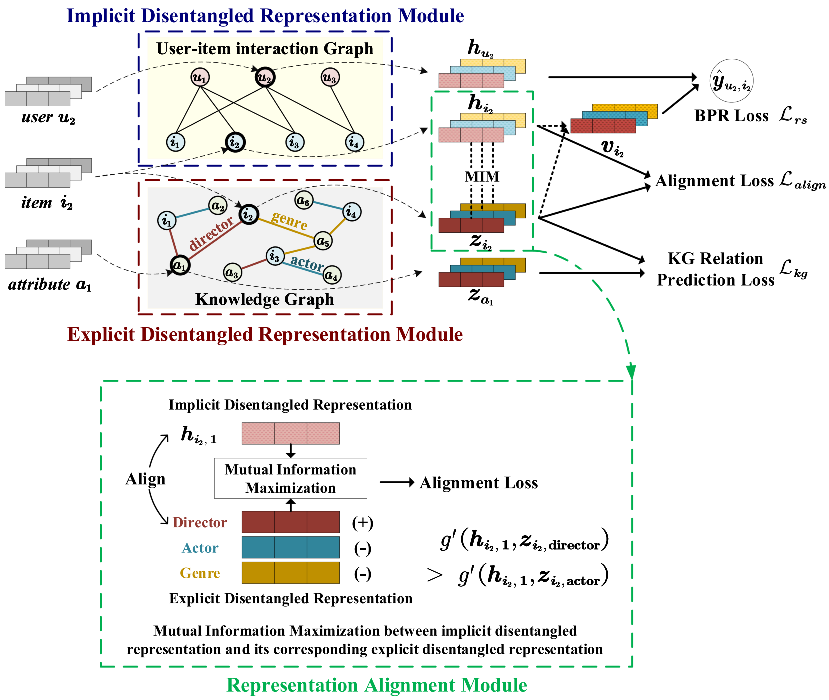


Fig. 1. The overview of the proposed KDR approach, consisting of three modules: implicit disentangled representation module, explicit disentangled representation module, and representation alignment module. Best viewed in color.

learned function will predict the score $\hat{y}_{u,i}$ that user u would interact with item, i by taking the learned user disentangled representation $\boldsymbol{v}_u$ and item, disentangled representation $\boldsymbol{v}_i$ as inputs. The recommender system ranks these predicted scores between a user and candidate items, and then selects the top-N items to recommend.

3 THE PROPOSED KDR APPROACH

In this section, we present the proposed KDR learning approach, to the top-N recommendation task. We first give an overview of the proposed KDR approach, and then describe each module in detail, and finally present the discussion and analysis.

Figure 1 presents an overview of the proposed KDR approach, consisting of three major modules: implicit disentangled representation module, knowledge guidance module, and representation alignment module.

In the implicit disentangled representation module, we use **disentangled graph convolutional network (DisenGCN)** [31] to learn *implicit disentangled representations* from the user-item interaction graph to model user preferences and item, characteristics. However, such implicit disentangled representations lack of interpretability and the learned representations for inactive users and items may not be sufficient. In the explicit disentangled representation module, we use a **multi-representation Relational Graph Convolution Network (mRGCN)** structure to learn *explicit disentangled representations* from the KG. Finally, in the representation module, we propose a mutual information maximization strategy to align the disentangled aspects between implicit and explicit disentangled representations obtained from the first two modules, which guides implicit disentangled representation learning via structural KG semantics.

3.1 Implicit Disentangled Representation Module

Intuitively, user-item interaction is often affected by many factors, such as *director*, *actor*, and *genre* for watching movies, which are hidden or unobserved through the interaction behaviors. In this part, we adopt the DisenGCN [31] to discover such factor information implicitly contained in the user-item interaction graph $\mathcal{G}_{rs}$. The output of this module is multi-aspect representations for users and items, called *implicit disentangled representations*.

For notation convenience, we introduce two node placeholders of s and t on a graph, which can be a user or an item, in our setting. DisenGCN [31] designs a neighbor routing mechanism to dynamically identify the latent factors for the interaction (i.e., edge) between two nodes, and accordingly extracts the information under these different factors. Thus, DisenGCN can learn the disentangled node representations for a graph. The key element of DisenGCN is the DisenConv layer $f(\cdot)$ that updates a disentangled representation $\boldsymbol{h}_s^{(l)}$ for a node s, given the previous representations at the $(l-1)$th layer of this node s and its neighbors $\{t\}$:

$$\boldsymbol{h}_s^{(l)} = f\left(\boldsymbol{h}_s^{(l-1)}, \{\boldsymbol{h}_t^{(l-1)} : (s,t) \in \mathcal{G}_{rs}\}\right). \tag{1}$$

The output $\boldsymbol{h}_s^{(l)} = [\boldsymbol{h}_{s,1}^{(l)}, \boldsymbol{h}_{s,2}^{(l)}, \ldots, \boldsymbol{h}_{s,K}^{(l)}] \in \mathbb{R}^{K \times d}$ can be regarded as the disentangled representations of node s under K aspects at the lth layer, where $\boldsymbol{h}_{s,k}^{(l)} \in \mathbb{R}^d$ denotes the representation of node s in the kth aspect.

To learn the disentangled representations, DisenGCN needs to calculate the probability $p_{t,k}$ that node s reaches neighbor t due to the kth factor. The probabilities $\{p_{t,k}\}$ satisfy: $\forall k \in [1,K], p_{t,k} \geq 0$ and $\sum_{k=1}^{K} p_{t,k} = 1$. In a reverse direction, $p_{t,k}^{(l)}$ reflects the contribution probability that neighbor t constructs the representation $\boldsymbol{h}_{s,k}^{(l)}$. Thus the neighbor routing mechanism will alternatively construct $\boldsymbol{h}_{s,k}^{(l)}$ and infer $p_{t,k}^{(l)}$ as follows:

$$\boldsymbol{h}_{s,k}^{(l)} = \frac{\boldsymbol{h}_{s,k}^{(l-1)} + \sum_{t:(s,t)\in\mathcal{G}_{rs}} p_{t,k}^{(l)} \boldsymbol{h}_{t,k}^{(l-1)}}{\|\boldsymbol{h}_{s,k}^{(l-1)} + \sum_{t:(s,t)\in\mathcal{G}_{rs}} p_{t,k}^{(l)} \boldsymbol{h}_{t,k}^{(l-1)}\|_2}, \tag{2}$$

$$p_{t,k}^{(l)} = \frac{\exp(\boldsymbol{h}_{t,k}^{(l-1)} \cdot \boldsymbol{h}_{s,k}^{(l)}/\tau)}{\sum_{k=1}^{K} \exp(\boldsymbol{h}_{t,k}^{(l-1)} \cdot \boldsymbol{h}_{s,k}^{(l)}/\tau)}, \tag{3}$$

where τ is a hyper-parameter that controls the hardness of the assignment. The final $\boldsymbol{h}_s^{(l)}$ is the disentangled representations for node s at the lth layer. The DisenConv layer iteratively calculates the Equation (2) and Equation (3) by initializing:

$$p_{t,k}^{(l)} \propto \exp\left(\boldsymbol{h}_{t,k}^{(l-1)} \cdot \boldsymbol{h}_{s,k}^{(l-1)}/\tau\right). \tag{4}$$

The vector $\boldsymbol{h}_s^{(l)}$ is the disentangled representations for node s at the lth layer.

For user u and item, i in $\mathcal{G}_{rs}$, we use the initial embedding $\boldsymbol{v}_u$ and $\boldsymbol{v}_i$ as the input for the first DisenConv layer, i.e., $\boldsymbol{h}_u^{(0)} = \boldsymbol{v}_u$ and $\boldsymbol{h}_i^{(0)} = \boldsymbol{v}_i$. After L DisenConv layers, we obtain the implicit disentangled representations at the Lth layer: $\boldsymbol{h}_u = \boldsymbol{h}_u^{(L)}$ and $\boldsymbol{h}_i = \boldsymbol{h}_i^{(L)}$. Via the above neighbor routing mechanism, we can learn the implicit disentangled representations for users and items based on the user-item interaction graph $\mathcal{G}_{rs}$.

3.2 Explicit Disentangled Representation Module

In contrast to the above implicit learning way, we further utilize KG data to derive interpretable disentangled representations, since KG contains structural attribute information of the items in recommender system, such as *director*, *actor,* and *genre.* To effectively utilize KG data, we design a mRGCN model to learn multi-aspect representations (called *explicit disentangled representations*) for items from the KG $\mathcal{G}_{kg}$. We also design a relation prediction task to refine the disentangled representation learning.

In the original RGCN [37], each node has only one representation. While, in our task, we need to learn multiple-aspect representations for each node in KG. For this purpose, we propose mRGCN by incorporating a mRGCN layer $g(\cdot)$. mRGCN updates multiple-aspect representation $z_s^{(l)}$ for a node s given its neighbors $\{t\}$, their previous representations at the $(l-1)$th layer, and the relation links to its neighbors:

$$z_s^{(l)} = g\left(z_s^{(l-1)}, \{z_t^{(l-1)} : (s,r,t) \in \mathcal{G}_{kg}, r \in \mathcal{R}\}\right), \tag{5}$$

where again, s and t are the placeholders denoting any node on KG, and $z_s^{(l)}$ is the learned node representation for s.

The output $z_s^{(l)} = [z_{s,r_1}^{(l)}, z_{s,r_2}^{(l)}, \ldots, z_{s,r_K}^{(l)}] \in \mathbb{R}^{K \times d}$ can be regarded as the multiple-aspect representation of node s under K relations at the lth layer, where $z_{s,r_k}^{(l)} \in \mathbb{R}^d$ denotes the representation of node s under the relation r_k. Specifically, for $g(\cdot)$, we define the following propagation method to calculate the forward-pass update of nodes in a multi-representation relational graph:

$$z_{s,r}^{(l)} = \sigma\left(\sum_{t \in \mathcal{N}_s^r} \frac{1}{c_{s,r}} W_r^{(l-1)} z_{t,r}^{(l-1)} + W_0^{(l-1)} z_{s,r}^{(l-1)}\right), \tag{6}$$

where $\mathcal{N}_s^r$ denotes the set of neighbors of node s under relation $r \in \mathcal{R}$, and $c_{s,r} = |\mathcal{N}_s^r|$ is a normalization term, and $\sigma(\cdot)$ is the sigmoid function.

For item, entity i and attribute entity a in $\mathcal{G}_{kg}$, we use the initial embedding $\boldsymbol{v}_i$ and $\boldsymbol{v}_a$ as the input for the first mRGCN layer, so we have $z_i^{(0)} = \boldsymbol{v}_i$ and $z_a^{(0)} = \boldsymbol{v}_a$. We iteratively stack L layers to obtain the multi-aspect representations for items and attributes in KG: $z_i = z_i^{(L)}$ and $z_a = z_a^{(L)}$.

Although the extended mRGCN model incorporates relation-specific node embeddings, the update of each triple only involves the entity embeddings *w.r.t.* some specific relations. For a given node, the overall correlation between different vectors has not been considered. Hence, we further design an auxiliary loss to enhance the KG multi-representation learning. The main idea, is to utilize the learned item, representation z_i and attribute representation z_a to predict their relation r. This can be easily implemented via the cross-entropy loss:

$$\mathcal{L}_{kg} = \sum_{\langle i,r,a \rangle \in \mathcal{G}_{kg}} \text{CrossEnt}\Big(\boldsymbol{l}_r, \text{softmax}([z_{i,1} \cdot z_{a,1}, \ldots, z_{i,K} \cdot z_{a,K}])\Big), \tag{7}$$

where $\boldsymbol{l}_r \in \mathbb{R}^K$ is the ground-truth relation label in the form of one-hot vectors, and softmax$([z_{i,1} \cdot z_{a,1}, \ldots, z_{i,K} \cdot z_{a,K}])$ indicates the predictive distribution over the selected relation set from a softmax function taking as input the vector $[z_{i,1} \cdot z_{a,1}, \ldots, z_{i,K} \cdot z_{a,K}] \in \mathbb{R}^K$.

3.3 Representation Alignment Module

So far we have discussed how to obtain implicit disentangled representations $\{\boldsymbol{h}_u\}_{u \in \mathcal{U}}$ and $\{\boldsymbol{h}_i\}_{i \in \mathcal{I}}$ from the user-item interaction graph $\mathcal{G}_{rs}$, and obtain explicit disentangled representations $\{z_i\}_{i \in \mathcal{I}}$ from the KG $\mathcal{G}_{kg}$.

As discussed above, we aim to utilize the KG information to guide the disentangled representation learning. The main idea, is to align the implicit disentangled representations with explicit disentangled representations that encode structural semantics from KG. We enforce an alignment of the disentangled factors between the implicit and explicit spaces, so that the implicit and explicit disentangled representations of an item, are driven to be close. In this way, the implicit disentangled representations will become more interpretable as guided by structural semantics from KG.

To model such an idea, we are inspired by recent progress in contrastive learning [19, 33, 39], and design a loss function based on contrastive learning that maximizes the mutual information between explicit and implicit disentangled representations. Formally, let $I(\boldsymbol{x}, \boldsymbol{y})$ denote the mutual information between two variables $\boldsymbol{x}$ and $\boldsymbol{y}$, which can be understood as how much knowing $\boldsymbol{x}$ reduces the uncertainty in $\boldsymbol{y}$. Recall that each item, i is associated with two types of representations at each disentangled aspect, either implicit ($\boldsymbol{h}_{i,k}$) or explicit ($\boldsymbol{z}_{i,r_k}$), so we would like to maximize the mutual information between $\boldsymbol{h}_i$ and $\boldsymbol{z}_i$, which drives $\boldsymbol{h}_{i,k}$ and $\boldsymbol{z}_{i,r_k}$ to be similar. Maximizing mutual information directly is usually intractable. Following [19, 33], we adopt the Jensen-Shannon mutual information estimator for the kth disentangled aspect:

$$\begin{aligned} & I_k^{(\mathrm{JS})}\left(\{\boldsymbol{h}_{i,k}\}_{i\in\mathcal{I}}, \{\boldsymbol{z}_{i,r_k}\}_{i\in\mathcal{I}}\right) \\ = \; & \mathbb{E}_{i\in\mathcal{I}}\left[-\mathrm{sp}(-g(\boldsymbol{z}_{i,r_k}, \boldsymbol{h}_{i,k}))\right] - \mathbb{E}_{i\in\mathcal{I}, k'\neq k}\left[\mathrm{sp}(g(\boldsymbol{z}_{i,r_{k'}}, \boldsymbol{h}_{i,k}))\right], \end{aligned} \tag{8}$$

where we compute the expectation based on the (uniform) item, distribution or aspect distribution, $\mathrm{sp}(x) = \log(1 + e^x)$ is the softplus function. The function $g(\cdot, \cdot)$ is implemented with inner product:

$$g\left(\boldsymbol{z}_{i,k}, \boldsymbol{h}_{i,k}\right) = \boldsymbol{z}_{i,k} \cdot \boldsymbol{h}_{i,k}. \tag{9}$$

Considering that there are K disentangled aspects, we accumulate the mutual information maximization loss over these K aspects, and the alignment loss is computed by minimizing the negative sum as follows:

$$\mathcal{L}_{align} = -\sum_{k=1}^{K} I_k^{(\mathrm{JS})}(\{\boldsymbol{h}_{i,k}\}_{i\in\mathcal{I}}, \{\boldsymbol{z}_{i,r_k}\}_{i\in\mathcal{I}}). \tag{10}$$

After disentangled representation alignment, we can use these representations to derive final disentangled representations for item, recommendations task. For user u, we directly use the implicit disentangled representation $\boldsymbol{h}_u$ as the final user representation $\boldsymbol{v}_u \in \mathbb{R}^{K\times d}$:

$$\boldsymbol{v}_u = \boldsymbol{h}_u. \tag{11}$$

For item, i, we sum the implicit and explicit disentangled representations as the final item, representation $\boldsymbol{v}_i \in \mathbb{R}^{K\times d}$:

$$\boldsymbol{v}_i = \boldsymbol{h}_i + \boldsymbol{z}_i. \tag{12}$$

In this way, these two kinds of disentangled representations complement each other, and enhance the item, representations.

3.4 Learning and Discussion

In this part, we introduce the learning algorithm, and present some discussion and analysis. The overall learning procedure is shown in Algorithm 1.

3.4.1 Learning. With the learned disentangled representations, we can make recommendation prediction as follows:

$$\hat{y}_{u,i} = \boldsymbol{v}_u \cdot \boldsymbol{v}_i, \tag{13}$$

where $\hat{y}_{u,i}$ is the matching score between user u and item, i.

To optimize the recommendation performance, we opt for the **Bayesian Personalized Ranking** (**BPR**) loss [34]. Specifically, it assumes that the observed interactions, which reflects user preferences, should be assigned higher prediction scores than unobserved ones:

$$\mathcal{L}_{rs} = -\sum_{(u,i^+,i^-)\in\mathcal{D}} \log\sigma\left(\hat{y}_{u,i^+} - \hat{y}_{u,i^-}\right), \tag{14}$$

where $\mathcal{D}$ denotes the training set, i^+ and i^- denote the observed or unobserved items in the interaction records of user u, respectively.

The final objective function is to minimize losses in Equations (7), (10), and (14) jointly, as follows:

$$\mathcal{L} = \mathcal{L}_{rs} + \lambda_1\mathcal{L}_{kg} + \lambda_2\mathcal{L}_{align} + \lambda_3\|\Theta\|_2^2, \tag{15}$$

where λ_1, λ_2, and λ_3 are balancing hyper-parameters, and Θ denotes all the model parameters. All three modules are trained together by minimizing loss $\mathcal{L}$ from scratch.

3.4.2 Model Analysis. We first understand the difference between our approach and previous studies on disentangled representation methods in recommender systems [31, 53]. To make the comparison, we first rewrite Equation (13) over the possible aspects as:

$$\hat{y}_{u,i} \quad = \quad \sum_{k=1}^{K} \boldsymbol{v}_{u,k} \cdot \boldsymbol{v}_{i,k}. \tag{16}$$

As we can see, the final prediction is based on the overall match results over the K aspects. Actually, almost all disentangled representation methods in recommender systems [31, 53] adopt the similar prediction formula by combining the matching degrees in different aspects. Compared with prior studies, a significant difference is that our disentangled dimensions or aspects (i.e., $\boldsymbol{v}_{i,k}$) are more interpretable since they are aligned to KG relations. As shown in Equation (16), it is more intuitive to understand the recommendation results if the enumerated dimensions (indexed by k) are interpretable. While, the disentangled aspects in previous studies [31, 53] are usually difficult to understand, which will be illustrated in Section 4.4.1. Our approach enhances the interpretability of the recommendation algorithms. We have made important technical contributions by learning and aligning explicit and implicit disentangled representations. Besides, GNNs have been widely explored for knowledge-aware recommendation models [46, 51]. Our work can be considered as an important extension to these studies. Instead of encoding a single node representation, we learn knowledge-aware disentangled representations for recommender system. As shown in Equation (16), by using a multi-aspect representation, the learned node embeddings are easy to explain and understand: each disentangled factor corresponds to a specific relation in KG. The disentangled representations reflect user preference or item, characteristics in different aspects. Furthermore, the user-item match or interaction can be modeled in a more accurate way. Note that our focus is how to learn the disentangled representations of users and items, while it is easy to make extensions to the final prediction formula, e.g., learning a weighted combination.

3.4.3 Complexity Analysis. We discuss about the time and space complexity in this part.

Time Complexity. In our approach, the most time-consuming modules are the first two modules. The time complexities for the two modules per iteration can be roughly estimated as $O(L\cdot E_{rs}\cdot K\cdot d)$ and $O(L\cdot E_{kg}\cdot K\cdot d^2)$, where L is the number of layers, E_{rs} is the number of edges on $\mathcal{G}_{rs}$, E_{kg} is the number of triples on $\mathcal{G}_{kg}$, K is the number of aspects, and d is the embedding size. Note that these two modules can set different layer numbers. A major increase on time cost lies in

ALGORITHM 1: The training algorithm for our proposed model.

Input: The user-item interaction graph $\mathcal{G}_{rs}$ and the knowledge graph $\mathcal{G}_{kg}$.
Output: Disentangled user representations $\{\boldsymbol{v}_u | u \in \mathcal{U}\}$, disentangled item, representations $\{\boldsymbol{v}_i | i \in \mathcal{I}\}$ and attribute entity representations $\{\boldsymbol{v}_a | a \in \mathcal{E}_a\}$.
Randomly initialize $\boldsymbol{v}_u$, $\boldsymbol{v}_i$ and $\boldsymbol{v}_a$ for each u in $\mathcal{U}$, i in $\mathcal{I}$ and a in $\mathcal{E}_a$.
for *each training iteration* **do**
  Sample a mini-batch of positive and negative interactions from $\mathcal{G}_{rs}$.
  Sample a mini-batch of relations and attribute entities from $\mathcal{G}_{kg}$.
  Compute the implicit disentangled representation $[\boldsymbol{h}_{u,1}, \ldots, \boldsymbol{h}_{u,K}]$ for users, and the implicit disentangled representation $[\boldsymbol{h}_{i,1}, \ldots, \boldsymbol{h}_{i,K}]$ for items according to Equation (1).
  Compute the explicit disentangled representation $[\boldsymbol{z}_{i,1}, \ldots, \boldsymbol{z}_{i,K}]$ for items according to Equation (5).
  Compute the relation prediction loss $\mathcal{L}_{kg}$ according to Equation (7).
  Compute the alignment loss $\mathcal{L}_{align}$ according to Equation (10).
  Compute the bayesian personalized ranking loss $\mathcal{L}_{rec}$ according to Equation (14).
  Compute the sum loss $\mathcal{L}$ according to Equation (15).
  Update $\boldsymbol{v}_u$, $\boldsymbol{v}_i$ and $\boldsymbol{v}_a$ according to $\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_u}$, $\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_i}$ and $\frac{\partial \mathcal{L}}{\partial \boldsymbol{v}_a}$ on the mini-batch by back-propagation.
end for

return $\{\boldsymbol{v}_u | u \in \mathcal{U}\}$, $\{\boldsymbol{v}_i | i \in \mathcal{I}\}$, $\{\boldsymbol{v}_a | a \in \mathcal{E}_a\}$.

the aspect number. We empirically find a small number of high-quality aspects can lead to a good performance, since our aspects align to KG relations or attributes. Another strategy to control time complexity is to use a small embedding size d. Take previous KG-aware recommendation methods [5, 51, 60] as a comparison. Assume the embedding size in these single-representation models is set to d'. We can proportionally set the embedding size of each disentangled representation as: $d = \frac{d'}{K}$. In this way, our model will have comparable (or even better) time complexities compared with previous KG-based recommendation methods. As will be shown in experiments, our model can achieve better performance and interpretability.

Space Complexity. The parameters mainly consist of two parts: embedding matrix and weight matrix in the mRGCN layer. The space complexities can be roughly estimated as $O(N \cdot K \cdot d + L \cdot K \cdot d^2)$, where N is the total number of users, items, and attribute entities, L is the number of layers, K is the number of disentangled aspects, and d is the embedding size of each disentangled aspect. Take previous single-representation KG recommendation methods [5, 51, 60] as a comparison: assume the embedding size in these single-representation models is set to d'. We can proportionally set the embedding size of each disentangled representation as: $d = \frac{d'}{K}$. In this way, our model has comparable space complexities with other single-representation KG recommendation methods. If a large number of relations is involved, we can derive each disentangled aspect embedding by a linear combination of a set of learnable relation embeddings. In this way, each disentangled aspect will correspond to a group of relation embeddings, which is also highly interpretable but with fewer parameters. Such a similar approach has been adopted by relation-based GNNs, such as RGCN [37] and CompGCN [42].

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the proposed approach from the following perspectives:

— How does the proposed approach KDR perform compared with the baseline methods?
— Can KDR learn disentangled representations with the help of KG?

Table 2. Statistics of the Datasets

	Music	Movie	Book
#User	15,864	6,040	20,314
#Item	28,378	3,238	29,692
#Interaction	697,818	995,987	560,263
#Entity	241,663	64,071	57,763
#Relation Type	10	12	10
#Triplet	618,850	277,838	220,574

— How does KDR perform under various settings such as different data sparsity levels and different hyper-parameters?

4.1 Experimental Setup

We first introduce the experimental setup, including adopted datasets, compared methods, and evaluation metrics.

4.1.1 Datasets. To evaluate the effectiveness of our method, we use three widely adopted real-world recommendation datasets, namely MovieLens-1m movie [13], LFM-1b music [36], and Amazon book [15]. These three datasets contain user-item interactions from different domains. The MovieLens dataset is collected by GroupLens Research from the MovieLens website, which is an online movie recommender system. It contains different datasets of varying sizes. And we use the MovieLens-1m, which is a commonly used benchmark dataset. It records about one million ratings from 6,000 users on 4,000 movies. The LFM-1b dataset is foremost intended for benchmarking in music recommendation. It is collected from Last.fm and contains more than one billion music listening events. Each listening event is characterized by artist, album, and track name and further includes a timestamp. Because it is very large, we only take the subset from the last year. The Amazon book dataset is collected from Amazon, which contains product ratings, reviews, and metadata. We use the version released in the year 2014. The product reviews are not used in our recommendation scenario, so we only keep the rating data. Following [35], for all datasets, unpopular items and inactive users are filtered out to ensure that all the items and users have at least 10 interaction records.

In our setting, both user-item interaction data and associated KG are required. We adopt Freebase [11] as the KG. Freebase stores facts by triples of the form $< head, relation, tail >$, and we use the last public version released in March 2015. Then we utilize the KB4Rec [64] dataset to align Freebase entities and items from the three user-item interaction datasets. After aligning, following [14] we use the aligned items as seeds, and generate the KG subgraph by performing breadth-first-search in each domain. Following [4, 14], we removed relations like $< film.director.film >$ which just reverses the head and tail compared to the relations $< film.film.directed\_by >$. Furthermore, to ensure the quality of KG, we removed relations that end up with non-freebase string, e.g., like $< film.film.rottentomatoes\_id >$ and filtered infrequent entities with fewer than three KG triples.

The statistics of these adopted datasets and associated KG are summarized in Table 2.

4.1.2 Baselines. We consider the following methods for performance comparison.

— **BPRMF** [34]: This method adopts matrix factorization techniques to learn latent representations for users and items and conduct inner product between them to predict an interaction. It's optimized by a pairwise ranking loss in the Bayesian approach.

- **NGCF** [52]: This method exploits the user-item interaction graph structure by propagating node embeddings on it. It achieves the target by leveraging high-order connectivities in the user-item interaction graph, which can better utilize the collaborative information between users and items.
- **LightGCN** [16]: This method removes the unnecessarily complicated design of GCNs for collaborative filtering. It consists of two essential components: light graph convolution and layer combination, which is a state-of-the-art GGN-based recommender model.
- **DisenGCN** [31]: This method proposes a GNN to learn disentangled node representations. It uses a neighborhood routing mechanism, which is capable of dynamically identifying the latent factor that may have caused the edge, between a node and one of its neighbors, and accordingly assigning the neighbor to a channel that extracts and convolutes features specific to that factor. We apply DisenGCN on user-item interaction graph for item, recommendations.
- **DGCF** [53]: This method utilizes the graph disentangling module to iteratively refine the intent-aware interaction graphs and factorial representations. It is a state-of-the-art method in learning disentangled representations for recommender systems.
- **CKE** [60]: This is an embedding-based method that can incorporate KG and other information such as corresponding images and texts to enrich the representation of items. In our implementation, we only incorporate KG information but not images and texts for a fair comparison.
- **KTUP** [5]: This is an embedding-based method which adopts the strategy of multi-task learning to jointly learn recommendation and KG-related tasks, with the goal of understanding the reasons that a user interacts with an item. Different from the disentangled representations in recommender system, this method utilizes an attention mechanism to combine all preferences into a single-vector representation.
- **KGAT** [51]: This method explores high-order connectivity with semantic relations in **collaborative knowledge graph** (**CKG**) for knowledge-aware recommendation. It designs the attentive embedding propagation layer, which adaptively propagates the embeddings from a node's neighbors to update the nodes' representation. It's a state-of-the-art knowledge-aware recommendation model.
- **RGCN** [37]: This method proposes a relational graph convolutional network for the knowledge graph completion task. It uses the relation-specific transformations to aggregate the information from KG. Here we transfer it to the recommendation task.

Our baselines have a good coverage over different categories, including traditional method BPRMF, Neural based method NCF, GNN based method NGCF, disentangle learning methods DisenGCN and DGCF, and knowledge-aware or hybrid methods CKE, KTUP, KGAT, and RGCN.

4.1.3 Parameter Settings. To ensure a fair comparison, we optimize all the methods with Adam optimizer, where the batch size is set to be 2,048. The default Xavier initializer is used to initialize the model parameters. We apply a grid search for other hyper-parameters: the learning rate is tuned among $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$, the coefficient of L_2 normalization is searched in $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$, and the dropout ratio is tuned in $\{0.1, 0.2, \ldots, 0.8\}$. For GNN-based models, we search the number of layers in $\{1, 2, 3\}$. For disentangle learning models, we search the number of disentangled aspects K in $\{1, 2, \ldots, |\mathcal{R}|\}$, where $|\mathcal{R}|$ is the number of relations in KG. The initial embedding size is fixed to be 96 for single-vector representation models. For multi-vector representation models, we keep the same *total* embedding size to give a fair comparison. To be specific, for multi-vector representation models, the dimensionality of each aspect representation is set to be $\lfloor 96/K \rfloor$.

4.1.4 Evaluation Metrics. To evaluate the performance, we adopt a variety of evaluation metrics widely used in previous studies [21, 46, 51], including Precision@K, Recall@K, and NDCG@K:

$$\text{Precision@}K = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[\omega(k) \in \mathcal{I}_u], \tag{17}$$

$$\text{Recall@}K = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{|\mathcal{I}_u|} \sum_{k=1}^{K} \mathbb{I}[\omega(k) \in \mathcal{I}_u], \tag{18}$$

$$\text{NDCG@}K = \frac{1}{|\mathcal{U}_{test}|} \sum_{u \in \mathcal{U}_{test}} \frac{1}{IDCG_u} \sum_{k=1}^{K} \frac{2^{\mathbb{I}[\omega(k) \in \mathcal{I}_u]} - 1}{\log_2 (k+1)}, \tag{19}$$

where $\mathcal{U}_{test}$ is the set of users in test set, $\mathbb{I}[\cdot]$ is an indicator function that returns 1 when the condition is true, $\omega(k)$ is the item, at rank k, and $\mathcal{I}_u$ is the set of held-out items that user interacted with. For all these metrics, the larger value, the better performance.

For each dataset, we randomly select 80% of interactions as the training set, and the remaining is equally divided into the validation set and test set. For each user in the test set, we treat all the items that this user has not interacted with as negative items. All the reported performance numbers are the averaged results over all the users in the test set.

4.2 Performance Comparison

Table 3 presents the overall comparison between our approach KDR and baselines on top-N recommendation. From it, we can have the following observations:

(1) Among the methods that only utilize user-item interaction data, GNN-based methods NGCF, LightGCN, DisenGCN, and DGCF show a better performance than traditional method BPRMF, as they are more effective to learn useful semantic characteristics encoded in user-item graph structures. Specially, DisenGCN and DGCF can learn implicit disentangled representations for both users and items based on user-item interaction graph. These results indicate that disentangled representations can improve the recommendation performance by decoupling complex information from user-item interaction. Interestingly, LightGCN shows competitive results compared with disentangled representation learning method DisenGCN and DGCF. LightGCN makes some specific simplification or optimization based on NGCF, which has been verified to have a better performance than NGCF [52]. In particular, LightGCN has achieved a better performance than the other baselines (including DisenGCN and DGCF) on the music dataset. We speculate that the recommendation scenario on music dataset is simpler than the two datasets, and the disentangled learning based methods do not gain much by learning underlying factors. In Section 4.3.3, the experiment results on parameter sensitivity will further verify this point.

(2) Among the category of methods that utilize both user-item interaction data and KG data, CKE incorporates the side information in KG to enrich the representations of items, KTUP adopts the strategy of multi-task learning to use KG, KGAT learns the representations of users and items by exploiting the structure of KG and RGCN aggregates the information from KG and user-item interaction graph by the relation-specific transformations. Table 3 shows that CKE, KTUP, and KGAT perform better than RGCN. Because RGCN is not originally designed for recommendation and thus fails to model user-item relationships properly. KTUP and KGAT perform better than CKE as they integrate knowledge information into the recommender system more effectively. Generally speaking, these KG-aware baselines improve the recommendation performance by leveraging KGs

Table 3. Performance Comparison of Different Recommendation Algorithms

Dataset	Method	P@10	P@20	R@10	R@20	NDCG@10	NDCG@20
Music	BPRMF	0.0851	0.0597	0.2329	0.3104	0.1894	0.2155
	NGCF	0.0854	0.0598	0.2383	0.3144	0.1935	0.2194
	LightGCN	<u>0.0910</u>	<u>0.0625</u>	<u>0.2438</u>	<u>0.3222</u>	0.2018	0.2273
	DisenGCN	0.0874	0.0578	0.2402	0.3009	<u>0.2033</u>	0.2235
	DGCF	0.0872	0.0618	0.2397	0.3196	0.1969	0.2241
	CKE	0.0863	0.0602	0.2393	0.3115	0.1971	0.2238
	KTUP	0.0875	0.0608	0.2405	0.3147	0.1992	0.2242
	KGAT	0.0876	0.0613	0.2421	0.3122	0.2024	<u>0.2274</u>
	RGCN	0.0861	0.0602	0.2387	0.3121	0.1962	0.2221
	KDR	**0.0940***	**0.0627***	**0.2575***	**0.3248***	**0.2169***	**0.2394***
Moive	BPRMF	0.1676	0.1392	0.1439	0.2245	0.2087	0.2212
	NGCF	0.1704	0.1420	0.1516	0.2360	0.2126	0.2271
	LightGCN	<u>0.1717</u>	0.1429	0.1519	0.2383	<u>0.2164</u>	<u>0.2310</u>
	DisenGCN	0.1713	0.1443	0.1496	0.2385	0.2124	0.2287
	DGCF	0.1714	<u>0.1446</u>	<u>0.1520</u>	<u>0.2403</u>	0.2148	<u>0.2310</u>
	CKE	0.1702	0.1427	0.1499	0.2304	0.2106	0.2203
	KTUP	0.1707	0.1408	0.1502	0.2337	0.2122	0.2267
	KGAT	0.1712	0.1416	0.1507	0.2341	0.2145	0.2279
	RGCN	0.1706	0.1416	0.1491	0.2320	0.2145	0.2279
	KDR	**0.1756***	**0.1462***	**0.1592***	**0.2478***	**0.2201***	**0.2362***
Book	BPRMF	0.0202	0.0166	0.0706	0.1162	0.0481	0.0625
	NGCF	0.0197	0.0157	0.0672	0.1138	0.0473	0.0583
	LightGCN	0.0221	0.0179	0.0779	0.1263	0.0526	0.0678
	DisenGCN	0.0228	0.0183	0.0817	0.1292	0.0550	0.0700
	DGCF	0.0228	0.0179	<u>0.0840</u>	<u>0.1298</u>	<u>0.0582</u>	<u>0.0727</u>
	CKE	0.0211	0.0178	0.0773	0.1208	0.0526	0.0664
	KTUP	0.0212	0.0178	0.0777	0.1237	0.0524	0.0674
	KGAT	<u>0.0229</u>	<u>0.0184</u>	0.0782	0.1244	0.0546	0.0693
	RGCN	0.0181	0.0145	0.0648	0.1017	0.0430	0.0548
	KDR	**0.0248***	**0.0193***	**0.0911***	**0.1380***	**0.0630***	**0.0780***

The best performances are marked bold and the second best ones are underlined. * indicates the statistical significance for $p < 0.01$ compared to the best baseline. Note that we have adopted the full ranking by considering all the items that a user has not interacted with.

for enhancing the data representations. This confirms that KG is useful for the recommendations task.

(3) Finally, we can see that the proposed approach KDR consistently performs better than all the baseline methods by a large margin. This advantage is brought by the fact that the proposed KDR jointly utilizes both the user-item interaction data and KG data to learn the disentangled representations in recommender systems. By incorporating structural information from KG, our approach can effectively guide and enhance the disentangled representation learning.

Table 4. Ablation Study of the Proposed Method

Dataset	Method	P@10	P@20	R@10	R@20	NDCG@10	NDCG@20
Music	KDR	**0.0940**	**0.0627**	**0.2575**	**0.3248**	**0.2169**	**0.2394**
	w/o R	0.0903	0.0602	0.2459	0.3102	0.2070	0.2283
	w/o A	0.0892	0.0594	0.2425	0.3059	0.2047	0.2256
	w/o KG	0.0874	0.0578	0.2402	0.3009	0.2033	0.2235
Movie	KDR	**0.1756**	**0.1462**	**0.1592**	**0.2478**	**0.2201**	**0.2362**
	w/o R	0.1710	0.1442	0.1515	0.2399	0.2152	0.2316
	w/o A	0.1714	0.1430	0.1513	0.2352	0.2140	0.2283
	w/o KG	0.1713	0.1443	0.1496	0.2385	0.2124	0.2287
Book	KDR	**0.0248**	**0.0193**	**0.0911**	**0.1380**	**0.0630**	**0.0780**
	w/o R	0.0240	0.0187	0.0874	0.1344	0.0602	0.0745
	w/o A	0.0235	0.0183	0.0863	0.1314	0.0589	0.0733
	w/o KG	0.0228	0.0183	0.0817	0.1292	0.0550	0.0700

The Best Performances are Marked Bold.

4.3 Further Analysis of KDR

As shown in Table 3, our proposed approach KDR shows a better overall performance than all the baselines. In this section, we conduct a series of experiments to further analyze and understand the proposed approach.

4.3.1 Ablation Study. The proposed approach KDR utilizes knowledge to guide disentangled representation learning through three modules. So first, we examine the performance of KDR's variants by removing each module from the full approach. To be specific, we consider the following variants of KDR for comparison:

— *w/o R*: This variant removes KG relation prediction loss (Equation (7)) in the explicit disentangled representation module.
— *w/o A*: This variant removes the alignment loss (Equation (10)) between implicit disentangled representations and explicit disentangled representations in the representation alignment module.
— *w/o KG*: This variant removes both the explicit disentangled representation module and representation alignment module, which degenerates to the DisenGCN baseline.

In Table 4, it can be observed that all the proposed techniques or modules are useful to improve the final performance. Especially, the performance of the variant w/o KG is the worst, showing that the information from KG is essential for the disentangled representation learning in recommender systems. KDR can learn high-quality disentangled representations by utilizing the rich structural attribute information from KG. Then the result of the variant w/o A shows the alignment loss in representation alignment module plays an important role in performance improvement. The alignment loss enforces the disentangled representations to be aligned with structural knowledge information. The result of the variant w/o R shows the relation prediction loss in explicit disentangled representation module is also important. The relation prediction loss enhances the utilization of knowledge information to learn the explicit disentangled representations. These ablation studies confirm that all these three modules of KDR are useful by utilizing knowledge from KG to learn better disentangled representations, and these modules together enable the proposed KDR to significantly improve the recommendation results.

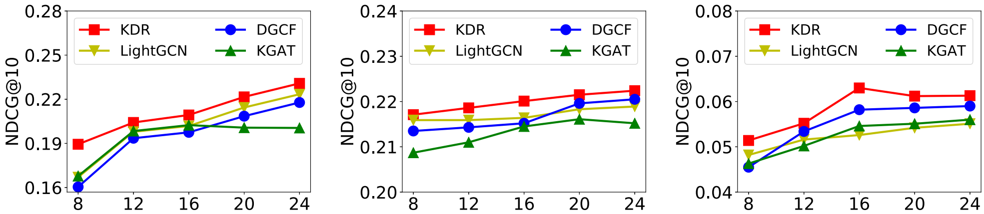
Table 5. Performance (NDCG@10) Comparison

Dataset	Method	*A*	*B*	*C*	*D*
Music	LightGCN	0.1956 (+8.2%)	0.1447 (+8.6%)	0.1904 (-3.2%)	0.2589 (+1.2%)
	DisenGCN	0.1982 (+6.8%)	0.1441 (+9.0%)	0.1810 (+1.8%)	0.2471 (+6.0%)
	DGCF	0.1993 (+6.2%)	0.1492 (+5.3%)	0.1802 (+2.3%)	0.2571 (+1.9%)
	KGAT	0.2021 (+4.7%)	0.1452 (+8.2%)	0.1792 (+2.8%)	0.2440 (+7.3%)
	KDR (w/o *R*)	0.2062 (+2.6%)	0.1567 (+0.3%)	0.1827 (+0.9%)	0.2572 (+1.8%)
	KDR (w/o *A*)	0.2025 (+4.5%)	0.1552 (+1.2%)	0.1826 (+0.9%)	0.2507 (+4.5%)
	KDR	**0.2116**	**0.1571**	**0.1843**	**0.2619**
Movie	LightGCN	0.1425 (+9.7%)	0.1673 (+2.2%)	0.2112 (+0.9%)	0.3287 (+0.7%)
	DisenGCN	0.1487 (+5.1%)	0.1680 (+1.7%)	0.2084 (+2.2%)	0.3264 (+1.4%)
	DGCF	0.1443 (+8.3%)	0.1667 (+2.5%)	0.2097 (+1.6%)	0.3278 (+1.0%)
	KGAT	0.1496 (+4.5%)	0.1679 (+1.8%)	0.2103 (+1.3%)	0.3249 (+1.9%)
	KDR (w/o *R*)	0.1532 (+2.0%)	0.1699 (+0.6%)	0.2092 (+1.8%)	0.3251 (+1.8%)
	KDR (w/o *A*)	0.1523 (+2.6%)	0.1661 (+2.9%)	0.2099 (+1.5%)	0.3250 (+1.8%)
	KDR	**0.1563**	**0.1709**	**0.2130**	**0.3310**
Book	LightGCN	0.0511 (+24.5%)	0.0452 (+11.5%)	0.0587 (+1.2%)	0.0922 (-4.8%)
	DisenGCN	0.0513 (+24.0%)	0.0462 (+9.1%)	0.0582 (+2.1%)	0.0864 (+1.6%)
	DGCF	0.0527 (+20.7%)	0.0475 (+6.1%)	0.0585 (+1.5%)	0.0870 (+0.9%)
	KGAT	0.0538 (+18.2%)	0.0482 (+4.6%)	0.0554 (+7.2%)	0.0847 (+3.7%)
	KDR (w/o *R*)	0.0612 (+3.9%)	0.0498 (+1.2%)	0.0585 (+1.5%)	0.0852 (+3.1%)
	KDR (w/o *A*)	0.0599 (+6.2%)	0.0482 (+4.6%)	0.0583 (+1.9%)	0.0861 (+2.0%)
	KDR	**0.0636**	**0.0504**	**0.0594**	**0.0878**

w.r.t. sparsity level. *A* to *D* denote scenarios with a decreasing sparsity level, i.e., *A* is the most sparse one.

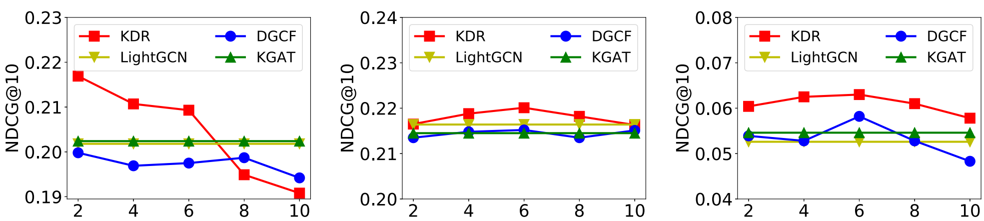
4.3.2 Performance Comparison w.r.t. Sparsity Levels. As discussed in Section 1, we incorporate KG into disentangled representation learning to solve two challenges: improving interpretability and resolving data sparsity issue. For interpretability, we will show it in Section 4.4.1. Here, we examine how the proposed method performs *w.r.t.* different sparsity levels. For this purpose, we divide the users into four groups according to the number of interaction records, and conduct experiments on those four split datasets. We summarize the experimental results in Table 5, in which dataset *A* contains the users with very few interaction records, while dataset *D* contains users with sufficient interaction records. From Table 5, we can observe that our approach KDR performs better than other baseline methods. Especially, on the book datasets, KDR yields a significant improvement under the sparse scenario (dataset *A*). These observations confirm that by incorporating information from KG, the proposed method KDR can learn better disentangled representations even for those inactive users and alleviate data sparsity issue. Besides, compared with the variants w/o *R* and w/o *A*, we find that the alignment loss designed in the representation alignment module is also helpful for the performance improvement under the sparsity setting, since KG information can be better used to guide the disentangled representation learning in this way.

4.3.3 Parameter Sensitivity. In this part, we further investigate the influence of model parameters on the performance to verify its robustness. For simplicity, we incorporate the three strongest baselines LightGCN, DGCF, and KGAT for comparisons.



(a) Music (b) Movie (c) Book

Fig. 2. Parameter sensitivity study on the each aspect embedding size. Note that LightGCN and KGAT have not included aspect embeddings, which are set to have the same total embedding size in comparison.



(a) Music (b) Movie (c) Book

Fig. 3. Parameter sensitivity study on the number of disentangled aspects. Note that LightGCN and KGAT has not included aspect embeddings, which are set to have the same total embedding size in comparison.

Figure 2 reports the results of varying the embedding size of disentangled aspect representations. We fix the number of disentangled aspects K unchanged, i.e., $K = 6$. It can be seen that by increasing the embedding size within a certain range, the performances of all the methods are improved, while the proposed approach KDR is consistently better than the baselines, indicating the stability of the proposed method.

Figure 3 shows the results for different numbers of disentangled aspects. For a fair comparison, we fix the total dimensionality unchanged, i.e., 96. For multi-vector representation models, the dimensionality of each aspect representation is set to be $\lfloor 96/K \rfloor$. The performance of KGAT and LightGCN remains fixed, since it does not involve such a parameter. For movie and book dataset, we can observe that by increasing the number of disentangled within a certain range, the performance of the multi-aspect representation models KDR and DGCF improves, due to the advantage of more disentangled aspect information. The performance drops slightly when the number of disentangled aspects continues to increase. This may be caused by the fact that the embedding size of each disentangled representation becomes smaller when the number of aspects increases (as we keep the total dimensionality unchanged), and it is difficult to accurately model disentangled aspect information. Another interesting phenomenon is that the disentangled learning methods KDR and DGCF perform best when considering two disentangled aspects on the music dataset. A possible reason is that music recommendation in this dataset is a much simpler scenario, and a large number of disentangled factors may deteriorate the performance.

4.4 Qualitative Analysis

In previous experiments, we have shown that the proposed approach KDR can improve the recommendation performance and address the data sparsity issue. Here, we conduct qualitative analysis to study the disentangled representations that are learned by our approach.

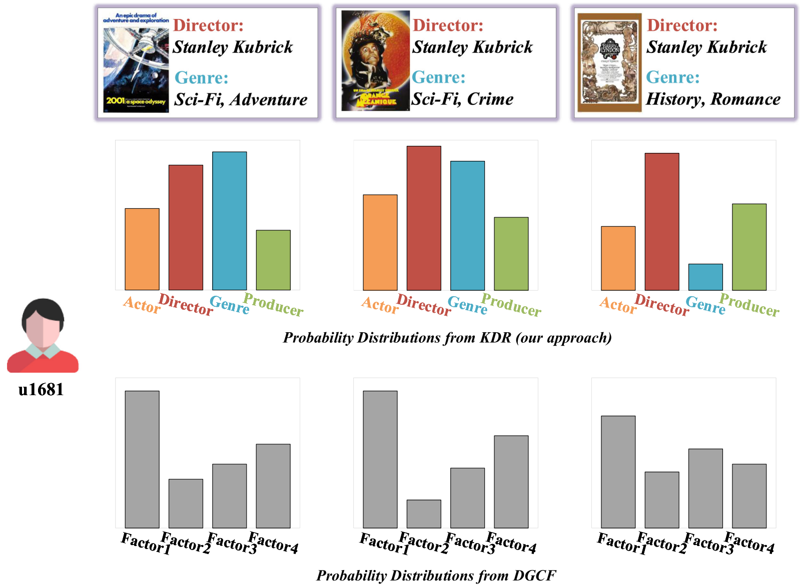


Fig. 4. A case study from the movie dataset. We include the results from DGCF as a comparison. For our results, we use different colors to denote different disentangled dimensions (Best viewed in color).

4.4.1 Case Study. In this part, we present a case study to qualitatively illustrate that the disentangled representations learned by our approach can improve the recommendation interpretability.

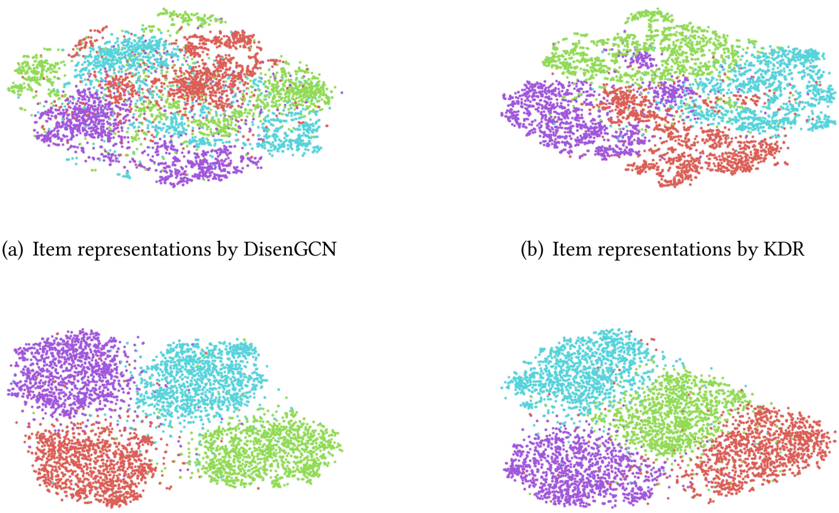
We randomly sample a user from the movie dataset as an illustrative example. This user has watched the presented three movies, namely "*2001: A Space Odyssey*", "*A Clockwork Orange*" and "*Barry Lyndon*", which are directed by the same director *Stanley Kubrick*. In our dataset, the three movies are associated with four kinds of attributes, including actor, director, genre, and producer.

As shown in Figure 4, the disentangled aspects by our approach are directly aligned with the four attributes, which are highly interpretable. In our approach, with the aligned disentangled dimensions, we can analyze how each factor contributes to user-item interaction. Specially, for each attribute, we compute the similarity between the disentangled aspect representations of the user and a movie, and normalize these attribute-specific similarities to a probability distribution over attributes. Such a distribution reflects the importance of different attributes on one specific user-item interaction. By looking into the results in Figure 4, two major influencing factors for the first two movies are *director* and *genre*, while the user has watched the third movie *Barry Lyndon* just because its director is his favorite (with few watching records in the genre of *History, Romance*).

Recall the baseline method DGCF can learn the disentangled aspects and compute the probability distribution over them (Equation (16)). Without knowledge guidance, although we can compute the probability distribution over different disentangled factors, it is usually difficult to explain the semantics of disentangled dimensions as shown at the bottom of Figure 4.

By making such a comparison, it shows that learned disentangled representations have clear meaning and provide good interpretability for the recommendation results.

4.4.2 Visualization of Disentangled Representations. To further confirm the usefulness of KG in disentangled representation learning, we use T-SNE [41] to visualize disentangled user and item,



(a) Item representations by DisenGCN (b) Item representations by KDR

(c) User representations by DisenGCN (d) User representations by KDR

Fig. 5. t-SNE visualization of disentangled representations on the movie dataset. Different colors represent the disentangled representations in different aspects.

representations learned by KDR and DisenGCN. Here, we incorporate DisenGCN for comparison, because it can be considered as a base model by removing all components and loss terms related to KG data. We visualize the results on the three datasets in Figure 5, Figure 6, and Figure 7, respectively.

As we can see, the user and item, representations learned by DisenGCN are not as good as those are learned by our approach, because there is no external knowledge to guide the disentangled learning process. Overall, for our approach KDR, the learned representations of different disentangled aspects are well separated and the representations of the same disentangled aspect are concentrated. An interesting observation is that the separation of item, representations is more clear than that of user representations. We speculate that users typically have more diverse, varying preferences, so that their representations are difficult to be separated.

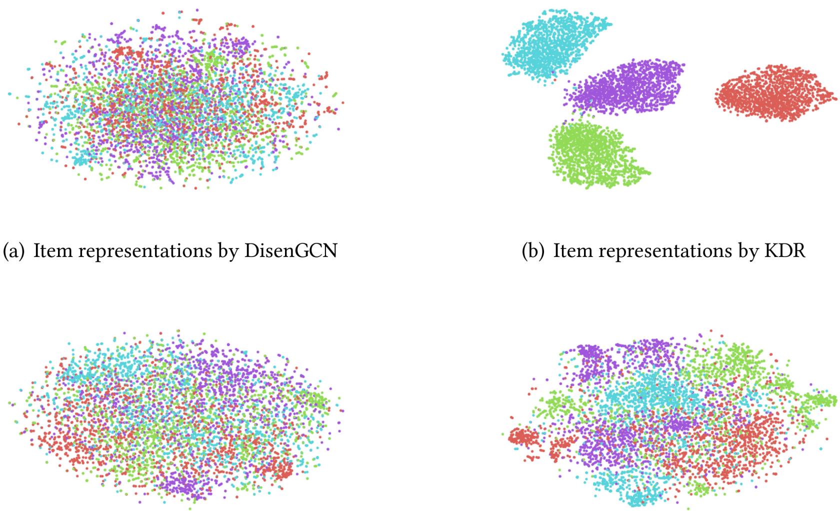
The visualization results on the three datasets demonstrate that the proposed approach KDR can utilize KG to help disentangled representation learning in recommender systems, and such disentangled representations can better capture user preferences and item, characteristics in different aspects.

5 RELATED WORK

In this section, we discuss existing work on item, recommendation, knowledge-aware recommendation, disentangled representation learning, and mutual information maximization, which are the most relevant topics with our work.

5.1 Item Recommendation

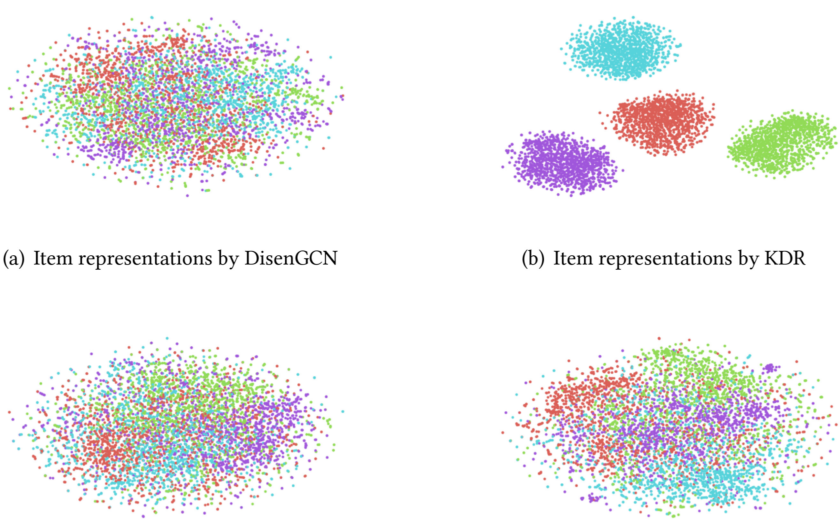
Item recommendation is an important task in recommender systems. It utilizes implicit feedback from user historical interactions to recommend a list of potentially interesting items to users. Early methods adopt matrix factorization techniques [25, 34] to learn latent representations for users and items, respectively, and conduct inner product between them to capture users' preference for



(a) Item representations by DisenGCN (b) Item representations by KDR

(c) User representations by DisenGCN (d) User representations by KDR

Fig. 6. t-SNE visualization of disentangled representations on the book dataset. Different colors represent the disentangled representations in different aspects.



(a) Item representations by DisenGCN (b) Item representations by KDR

(c) User representations by DisenGCN (d) User representations by KDR

Fig. 7. t-SNE visualization of disentangled representations on the music dataset. Different colors represent the disentangled representations in different aspects.

different items. With the success of deep learning, recent efforts [61, 65] focus on how to design effective neural architectures to capture the complex interactions between users and items. He et al. [17] replace the inner product in matrix factorization with a neural interaction architecture. Covington et al. [8] propose a deep neural architecture for Youtube video recommendation.

Xue et al. [57] use **Multi-Layer Perceptron** (**MLP**) to model the users and items, with the user-item interaction matrix as input. Afterwards, some studies adopt more advanced deep learning techniques to model users, items, and their interactions. Ebesu et al. [10] use memory network to capture the high-order relationship between users and items. Autoencoder and variational autoencoder are applied to collaborative filtering in server works [26, 55] as well. In addition to designing effective neural architectures, some methods try to apply advanced training strategies for item, recommendations, such as adversarial learning and reinforcement learning. Wang et al. [48] apply **generative adversarial networks** (**GAN**) to information retrieval and Chae et al. [6] apply GAN to collaborative filtering. Zhao et al. [66] use reinforcement learning to make full use of user negative feedback. More recently, GNN shows powerful modeling capabilities, which has been utilized in recommender systems to model high-order proximity among users and items [16, 52, 67]. Zheng et al. [67] exploits a spectral convolution operation on the user-item interaction graph to model high-order proximity in the spectral domain. Wang et al. [52] exploits the user-item interaction graph structure by propagating node embeddings on it, which better utilizes the collaborative information between users and items. He et al. [16] removes the unnecessarily complicated design of GNN for simplifying the recommendation network.

5.2 Knowledge-aware Recommendation

Recently, there is a surge of interest in leveraging KG information for improving recommender systems [12].

A typical approach is to apply KGE algorithms [50] to encode the KG into low-rank embedding, and then incorporate them into the recommendation systems [1, 5, 7, 22, 44, 47, 49, 58, 60]. Among these works, Zhang et al. [60] utilize various types of side information including item's structural knowledge to enrich the item, representations in recommender system. Wang et al. [45] propose RippleNet by utilizing memory networks to store and update the existing entity preference at each iteration. Huang et al. [22] propose to use memory networks to incorporate KG representations to capture the attribute-level user preferences and improve sequential recommendation. Yang et al. [58] introduce a GAN-based model to incorporate knowledge information for movie recommendation. Zhang et al. [1] combine user behaviors and item, knowledge information as a user-item KG and regard user behaviors (e.g., purchase, click) as one relation type between entities. Recently some methods [5, 7, 47] adopt multi-task learning to jointly learn recommendation tasks with other KG-related tasks. Cao et al. [5] jointly learn the task of recommendation and KG completion and the preference representations in the recommendation module are enriched by the KG. Meanwhile, Wang et al. [47] design a cross and compress unit to transfer knowledge and share regularization of items in the recommendation module and entities in the KG.

Besides, a number of studies leverage the connectivity patterns (relation-path or meta-path) of the entities in the KG for recommendation. The basic idea, is to capture high-level path semantics that enhance the relevance between two entities, and a typical approach is to encode the path information into low-dimensional vectors [40, 54, 56, 69]. Among these works, Wang et al. [54] construct the extracted relation path with both the entity embeddings and relation embeddings, and then encode them with an **long short-term memory (LSTM)** layer. Zhu et al. [69] incorporate the users' clicked history sequences and path connectivity between users and items with hierarchical attentional neural network for recommendation. Sun et al. [40] employ **recurrent neural network (RNN)** over the input knowledge entity embeddings along a meta-path to get the feature representation of the path between user-item pairs for recommendation. Besides, path-based semantics have been widely explored in **heterogeneous information network** (**HIN**) [21, 30, 38, 59, 62]. Their focus is to (usually manually) design several meta-paths that consist of a sequence of entity types, generate a number of paths by following these meta-paths, and evaluate the node similarity based on these paths. Among these works, Yu et al. [59] use the item-item meta-path based similarities as regularization terms in the matrix factorization framework. Luo et al. [30] consider the user-user similarity, item-item similarity, and user-item similarity for social-based recommendation. Shi et al. [38] use the ratings to build a weighted HIN to depict the path semantics through distinguishing different link attribute values. Furthermore, Zhao et al. [62] extends previous methods by adopting meta-graphs to better represent the high-level semantics of recommendations.

With the development of GNN, several studies construct a heterogeneous graph that jointly models user-item interaction and learn the representations of users or items based on a GNN-based architecture [46, 51, 63]. Wang et al. [51] model high-order connectivity with semantic relations in CKG using attentive embedding propagation layer. Wang et al. [46] apply a GNN to compute personalized item, embeddings and utilize a label smoothness mechanism to provide better inductive bias. Zhao et al. [63] exploit rich user-related behaviors in the graph for high-quality recommendation.

5.3 Disentangled Representation Learning

Disentangled representation learning aims to separate the underlying factors hidden in the data [3, 29]. Many early methods are from the field of computer vision [18, 20]. For example, Higgins et al. [18] propose to learn disentangled presentation for basic visual concepts with a constrained variational framework and Hsieh et al. [20] propose to learn the disentangled presentation for video prediction. Recently, disentangled representation learning on relational data has attracted increasing attention. Ma et al. [31] propose a disentangled graph convolutional network to learn node representations on graphs, which uses a neighbor routing mechanism to identify the latent factor and convolutes features specific to that factor. Liu et al. [28] propose a independence promoted graph disentangled network for graph representations, which enforces the disentangled factor more independent. There are also several efforts that apply disentangled representation learning to recommendation. Ma et al. [32] achieve macro disentanglement and micro disentanglement from user behaviors. Wang et al. [53] utilizes the graph disentangled module to iteratively refine the intent-aware interaction graphs and factorial representations for recommendation.

5.4 Mutual Information Maximization

Recently, contrastive learning has gained popularity because of its soaring performance on representation learning [27]. It learns the representation by comparing the data with positive and negative samples in the feature space. Mutual information maximization is a special branch of the contrastive learning. Mutual information targets at modeling the association between two variables, and is usually objected to be maximized. It has made important progress in several domains such as computer vision [2], nature language processing [24], graph learning [39, 43], and recommender system [68]. Deep InfoMax [19] is the first one to explicitly model mutual information through a contrastive learning task, which maximizes the mutual information between a local patch and its global context. Bachman et al. [2] propose to enhance the positive association between a local feature and its context by randomly sampling two different views of an image. In natural language processing, Kong et al. [24] propose to maximize mutual information between a global representation of a sentence and n-grams in it. In graph learning, Velickovic et al. [43] propose to maximize mutual information between patch representations and corresponding high-level summaries of graphs. Sun et al. [39] consider maximizing the mutual information between graph-level representation and substructures at different levels.

Different from the aforementioned studies, our work focuses on utilizing KG to guide disentangled representation learning in recommender systems, which can improve the effectiveness and interpretability of recommender systems. We make two important technical contributions.

First, we learn explicit disentangled representations by incorporating KG as the guidance signal for disentanglement. Second, we align explicit and implicit disentangled representations with mutual information maximization.

6 CONCLUSION

In this article, we proposed to utilize KG data to guide the disentangled representation learning in recommender systems. Our approach learned implicit and explicit disentangled representations from user-item interaction graph and KG, respectively, then adopted the mutual information maximization strategy to align the two kinds of disentangled representations, and finally fused them for item, recommendation. We designed a unified approach that effectively combined the above technical components. We conducted extensive experiments on three widely adopted real-world datasets from different domains.

Experimental results on three real-world datasets have shown that the proposed approach achieves the best performance compared to a number of competitive methods. Further experiments are conducted to analyze the usefulness of different modules in the proposed approach and demonstrate its stability under various scenarios. Especially, the disentangled representations in our approach are more interpretable by incorporating the KG semantics.

As future work, we will utilize other types of structural data to instruct disentangled representation learning. We will also consider using the current approach to improving other applications, e.g., causal inference for recommendation.

ACKNOWLEDGMENTS

The authors gratefully appreciate the anonymous reviewers for their valuable and detailed comments that greatly helped to improve the quality of this article.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.

[2] Philip Bachman, R. Devon Hjelm, and William Buchwalter. 2019. Learning representations by maximizing mutual information across views. In *Proceedings of the 32nd International Conference on Advances in Neural Information Processing Systems.* Vancouver, BC. 15509–15519.

[3] Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 8 (2013), 1798–1828.

[4] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of the 26th International Conference on Advances in Neural Information Processing Systems.* 2787–2795.

[5] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *Proceedings of the 28th International Conference on World Wide Web.* 151–161.

[6] Dong-Kyu Chae, Jin-Soo Kang, Sang-Wook Kim, and Jung-Tae Lee. 2018. CFGAN: A generic collaborative filtering framework based on generative adversarial networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management.* 137–146.

[7] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Jointly non-sampling learning for knowledge graph enhanced recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval.* 189–198.

[8] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems.* Boston, MA. 191–198.

[9] Paolo Cremonesi, Antonio Donatacci, Franca Garzotto, and Roberto Turrin. 2012. Decision-making in recommender systems: The role of user's goals and bounded resources. In *Proceedings of the 2nd Workshop on Human Decision Making in Recommender Systems.* 1–7.

[10] Travis Ebesu, Bin Shen, and Yi Fang. 2018. Collaborative memory network for recommendation systems. In *Proceedings of the 41st ACM SIGIR International Conference on Research and Development in Information Retrieval.* 515–524.

[11] Google. 2016. Freebase Data Dumps. (2016). Retrieved from https://developers.google.com/freebase/data.

[12] Qingyu Guo, Fuzhen Zhuang, Chuan Qin, Hengshu Zhu, Xing Xie, Hui Xiong, and Qing He. 2020. A survey on knowledge graph-based recommender systems. arXiv:2003.00911 Retrieved from https://arxiv.org/abs/2003.00911.

[13] F. Maxwell Harper and Joseph A. Konstan. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2016), 19:1–19:19.

[14] Gaole He, Junyi Li, Wayne Xin Zhao, Peiju Liu, and Ji-Rong Wen. 2020. Mining implicit entity preference from user-item interaction data for knowledge graph completion via adversarial learning. In Proceedings of *the Web Conference.* 740–751.

[15] Ruining He and Julian J. McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web.* 507–517.

[16] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd ACM SIGIR International Conference on Research and Development in Information Retrieval.* 639–648.

[17] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web.* 173–182.

[18] 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 *Proceedings of the 5th International Conference on Learning Representations.*

[19] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *Proceedings of the 7th International Conference on Learning Representations.*

[20] Jun-Ting Hsieh, Bingbin Liu, De-An Huang, Fei-Fei Li, and Juan Carlos Niebles. 2018. Learning to decompose and disentangle representations for video prediction. In *Proceedings of the 31st International Conference on Advances in Neural Information Processing Systems.* 515–524.

[21] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging meta-path based context for Top-N recommendation with A neural co-attention model. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 1531–1540.

[22] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of the 41st ACM SIGIR International Conference on Research and Development in Information Retrieval.* 505–514.

[23] Santosh Kabbur, Xia Ning, and George Karypis. 2013. FISM: Factored item similarity models for top-N recommender systems. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Chicago, IL. 659–667.

[24] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *8th International Conference on Learning Representations.*

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

[26] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web.* 689–698.

[27] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. arXiv:2006.08218 Retrieved from https://arxiv.org/abs/2006.08218.

[28] Yanbei Liu, Xiao Wang, Shu Wu, and Zhitao Xiao. 2020. Independence promoted graph disentangled networks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, the 32nd Innovative Applications of Artificial Intelligence Conference, the 10th AAAI Symposium on Educational Advances in Artificial Intelligence.* New York, NY. 4916–4923.

[29] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning.* California. 4114–4124.

[30] Chen Luo, Wei Pang, Zhe Wang, and Chenghua Lin. 2014. Hete-CF: Social-based collaborative filtering recommendation using heterogeneous relations. In *Proceedings of the 2014 IEEE International Conference on Data Mining.* 917–922.

[31] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. 2019. Disentangled graph convolutional networks. In *Proceedings of the 36th International Conference on Machine Learning.* 4212–4221.

[32] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. 2019. Learning disentangled representations for recommendation. In *Proceedings of the 32nd Annual Conference on Advances in Neural Information Processing Systems.* 5712–5723.

[33] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. 2016. f-GAN: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 29th Annual Conference on Advances in Neural Information Processing Systems*. 271–279.
[34] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. 452–461.
[35] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. 811–820.
[36] Markus Schedl. 2016. The LFM-1b dataset for music retrieval and recommendation. In *Proceedings of International Conference on Multimedia Retrieval*. 103–110.
[37] Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proceedings of the 15th International Conference on The Semantic Web*. 593–607.
[38] Chuan Shi, Zhiqiang Zhang, Ping Luo, Philip S. Yu, Yading Yue, and Bin Wu. 2015. Semantic path based personalized recommendation on weighted heterogeneous information networks. In *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*. 453–462.
[39] Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. 2020. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *Proceedings of the 8th International Conference on Learning Representations* .
[40] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 297–305.
[41] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.
[42] Shikhar Vashishth, Soumya Sanyal, Vikram Nitin, and Partha P. Talukdar. 2020. Composition-based multi-relational graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.
[43] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep graph infomax. In *Proceedings of the 7th International Conference on Learning Representations*.
[44] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Dvelopment in Information Retrieval*. 109–118.
[45] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. 417–426.
[46] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of International Conference on Knowledge Discovery & Data Mining*. 968–977.
[47] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-task feature learning for knowledge graph enhanced recommendation. In *Proceedings of International Conference on World Wide Web*. 2000–2010.
[48] Jun Wang, Lantao Yu, Weinan Zhang, Yu Gong, Yinghui Xu, Benyou Wang, Peng Zhang, and Dell Zhang. 2017. IRGAN: A minimax game for unifying generative and discriminative information retrieval models. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 515–524.
[49] 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 *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 209–218.
[50] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.
[51] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge graph attention network for recommendation. In *Proceedings of International Conference on Knowledge Discovery & Data Mining*. 950–958.
[52] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the SIGIR International Conference on Research and Development in Information Retrieval*. 165–174.
[53] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled representations for graph-based collaborative filtering. In *Proceedings of the 43rd ACM SIGIR International Conference on Research and Development in Information Retrieval*.
[54] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 5329–5336.

[55] Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*. 153–162.

[56] Yikun Xian, Zuohui Fu, S. Muthukrishnan, Gerard de Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd ACM SIGIR International Conference on Research and Development in Information Retrieval*. 285–294.

[57] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep matrix factorization models for recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 3203–3209.

[58] Deqing Yang, Zikai Guo, Ziyi Wang, Juyang Jiang, Yanghua Xiao, and Wei Wang. 2018. A knowledge-enhanced deep recommendation framework incorporating gan-based models. In *Proceedings of the IEEE International Conference on Data Mining*. 1368–1373.

[59] Xiao Yu, Xiang Ren, Quanquan Gu, Yizhou Sun, and Jiawei Han. 2013. Collaborative filtering with entity similarity regularization in heterogeneous information networks. In *IJCAI Workshop on Heterogeneous Information Network Analysis* .

[60] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 353–362.

[61] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys* 52, 1 (2019), 5:1–5:38.

[62] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-Graph based recommendation fusion over heterogeneous information networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 635–644.

[63] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. IntentGC: A scalable graph convolution framework fusing heterogeneous information for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2347–2357.

[64] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hong-Jian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. KB4Rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence* 1, 2 (2019), 121–136.

[65] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Kaiyuan Li, Yushuo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. arXiv:2011.01731 Retrieved from https://arxiv.org/abs/2011.01731.

[66] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1040–1048.

[67] Lei Zheng, Chun-Ta Lu, Fei Jiang, Jiawei Zhang, and Philip S. Yu. 2018. Spectral collaborative filtering. In *Proceedings of the 12th ACM Conference on Recommender Systems.* Vancouver, BC, Sole Pera, Michael D. Ekstrand, Xavier Amatriain, and John O'Donovan (Eds.). ACM, 311–319.

[68] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management.* 1893–1902.

[69] Qiannan Zhu, Xiaofei Zhou, Jia Wu, Jianlong Tan, and Li Guo. 2020. A knowledge-aware attentional reasoning network for recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, the 32nd Innovative Applications of Artificial Intelligence Conference, the 10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY*. 6999–7006.

Received December 2020; revised March 2021; accepted April 2021