Self-supervised Graph Disentangled Networks for Review-based Recommendation

Yuyang Ren[1] , **Haonan Zhang**[1] , **Qi Li**[1] , **Luoyi Fu**[1] , **Xinbing Wang**[1] and **Chenghu Zhou**[2]
[1]Shanghai Jiao Tong University
[2]Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences
{renyuyang, zhanghaonan, liqilcn, yiluofu, xwang8}@sjtu.edu.cn, zhouchsjtu@gmail.com

Abstract

User review data is considered as auxiliary information to alleviate the data sparsity problem and improve the quality of learned user/item or interaction representations in review-based recommender systems. However, existing methods usually model user-item interactions in a holistic manner and neglect the entanglement of the latent intents behind them, e.g., price, quality, or appearance. In this paper, we propose a Self-supervised Graph Disentangled Networks for review-based recommendation (SGDN), to separately model the user-item interactions based on the latent factors through the textual review data. To this end, we first model the distributions of interactions over latent factors from both semantic information in review data and structural information in user-item graph data, forming several factor graphs. Then a factorized message passing mechanism is designed to learn disentangled user/item and interaction representations on the factor graphs. Finally, we set an intent-aware contrastive learning task to alleviate the sparsity issue and encourage disentanglement. Empirical results over five benchmark datasets validate the superiority of SGDN over the state-of-the-art methods and the interpretability of learned intent factors.

1 Introduction

Review-based recommendation aims to alleviate the data sparsity problem [Mao *et al.*, 2016] in collaborative filtering methods [Koren *et al.*, 2022] by using the reviews of users to items as auxiliary information. Textual reviews contain useful semantic information that can be associated with the basis of users for their ratings, thereby leading to several investigations [Zheng *et al.*, 2017; Chen *et al.*, 2018] aimed to evaluate these user reviews to improve user preference modeling and rating predictions.

Early review-based recommendation methods usually employ deep neural networks such as Convolutional Neural Networks to model the review data for recommendation [Zheng *et al.*, 2017; Catherine and Cohen, 2017]. In addition, motivated by the attention mechanism [Vaswani *et al.*, 2017], many attention-based methods [Chen *et al.*, 2018; Wu *et al.*, 2019a] are proposed to identify the different importance of components, such as sentences, reviews, and users/items for better recommendation. More recently, the success of graph neural works [Kipf and Welling, 2016] in modeling the graph data also inspires its application in review-based recommender systems [Shuai *et al.*, 2022].

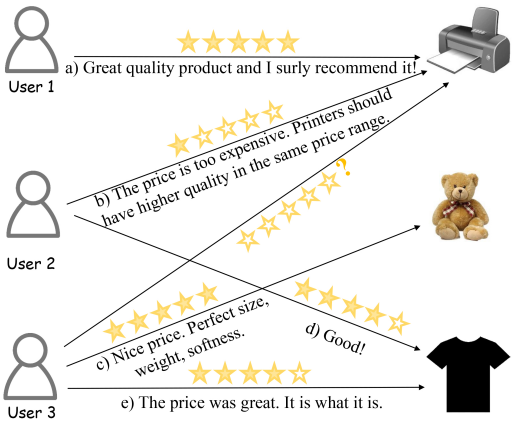


Figure 1: An example of user-item rating graph.

Despite their success, existing methods typically employ a holistic approach to leverage review data for user preference or user-item interaction modeling, i.e., either aggregating user/item reviews for user/item embedding learning [Chen *et al.*, 2018; Wu *et al.*, 2019b] or approximating the review of each user-item interaction based on learned user/item embeddings [Catherine and Cohen, 2017; Sun *et al.*, 2020]. Recently proposed RGCL [Shuai *et al.*, 2022] moves them forward by modeling the reviews as edge features in the user-item graph and incorporating them into the message passing process. However, user ratings of items are typically influenced by various complex latent intents, such as price, quality, appearance, etc. As shown in Figure 1, User 1 likes the printer because of its excellent quality, whereas User 2 dislikes it for its cheap cost performance. When it comes to predicting User 3's rating to the printer, we notice that User 3 is price sensitive based on his/her interactions with other items, so we anticipate a low rating score. Therefore, the complex latent factors underlying user-item interactions highlight a desire to disentangle these factors in the review-based

recommendation, which is still unexplored. As a result, the representations learned by existing methods contain a jumble of entangled factors, reducing interpretability and leading to suboptimal recommendation performance.

In this paper, we propose to learn disentangled user/item and interaction representations for better and more explainable review-based recommendation. To this end, we borrow the idea from disentangled representation learning (DRL) [Higgins *et al.*, 2016], which learns factorized representations to characterize the latent factors hidden in the data. Although introduced for some other recommendation tasks [Wang *et al.*, 2020; Ma *et al.*, 2020], DRL in review-based recommendation faces the following challenges.

- How to accurately identify the distribution of latent factors in the user-item interactions based on review and graph information and model the interactions at a finer granularity?
- How to design a proper self-supervised task based on the learned interaction representations to alleviate the sparsity issue and encourage disentanglement?

To tackle these challenges, we propose a novel Self-supervised Graph Disentangled Network (SGDN) for review-based recommendation. In particular, we first design a disentangled graph learning module equipped with graph disentangling and factorized message passing mechanisms. The former models the distribution of latent factors in each user-item interaction jointly from semantic information in the review and structure information in the user-item graph. The latter characterizes the user preferences from various aspects based on the generated factor graphs by accumulating factor-relevant information from neighborhoods. Furthermore, we present an intent-aware contrastive learning (CL) task that can alleviate the sparsity issue. Specifically, we dynamically select positive and negative samples for each interaction based on the learned intent distributions and maximize the agreement between positive pairs compared to negative ones. In comparison to existing methods, SGDN learns disentangled representations for users, items, and interactions, allowing it to investigate the meaning of each latent factor, resulting in greater explainability for predicting user ratings.

Our contributions can be mainly summarized as:

- Based on the review and graph information in review-based recommendation scenario, we make the first attempt to learn disentangled representations for users/items and interactions at a finer granularity.
- We propose a novel SGDN framework based on a contrastive learning task that alleviates the data sparsity issue as well as encourages disentanglement.
- We conduct extensive experiments on five datasets to validate the effectiveness and interpretability of SGDN.

2 Related Work

With the development of deep learning, many advanced text methods are used to extract semantic information from the review data. For instance, DeepCoNN [Zheng *et al.*, 2017] models user behaviors and item properties from review data using two parallel networks. TransNet [Catherine and Cohen, 2017] extends DeepCoNN by adding an additional layer for learning the target review features. In addition, inspired by attention mechanism [Vaswani *et al.*, 2017], DAML [Liu *et al.*, 2019] used the local and mutual attention of CNN to learn the user/item and interaction representations. DRRNN [Xi *et al.*, 2021] uses both target ratings and reviews for backpropagation to retain more semantic review information.

Recently, inspired by the success of Graph neural networks (GNNs) [Kipf and Welling, 2016], RMG [Wu *et al.*, 2019b] applied a three-level attention network to learn representations of sentence, review, and user/item and a graph attention network to model interactions. RGCL [Shuai *et al.*, 2022] incorporated review information as edge features into user/item embedding learning and designed two contrastive learning tasks as additional self-supervised signals. Summarizing existing review-based recommendation methods, they do not consider disentangling the hidden factors of user ratings, and the problem of data sparsity still exists.

Meanwhile, many contrastive learning (CL) paradigms have been designed to alleviate the data sparsity issue in recommender systems. For instance, SGL [Wu *et al.*, 2021] utilized an auxiliary GCL task to enhance user/item representation learning via self-discrimination. HCCF [Xia *et al.*, 2022b] conducted cross-view contrastive learning between the explicit interaction graph and the learned implicit hypergraph structure. While these works simply regard the same data instance and others as positive and negative samples, respectively, our intent-aware CL module selects proper positive and negative samples based on the learned intent distributions and further enhances the disentanglement.

Disentangled representation learning aims to learn factorized representations that reveal and disentangle the underlying latent factors hidden in the observed data [Ma *et al.*, 2019]. When it comes to recommendation, DGCF [Wang *et al.*, 2020] factorized the user-item graph into several intent-aware interaction graphs and iteratively update them based on user-item interaction. [Ma *et al.*, 2020] proposed a sequence-to-sequence training strategy based on latent self-supervision and disentanglement for sequential recommendation. Despite the promising performance, existing methods do not fit our task since they ignore the fruitful semantic information hidden in the review texts.

3 Problem Definition

In the task of review-based recommendation, we denote $\mathcal{U}$ ($|\mathcal{U}| = M$) as the user set and $\mathcal{I}$ ($|\mathcal{I}| = N$) as the item set. The rating record is formulated as a user-item rating matrix $R \in \mathbb{R}^{M \times N}$, where R_{ij} denotes the rating score of user i to item j and $\mathcal{R}$ denotes the set of all the possible ratings in the dataset (e.g., $\mathcal{R} = \{1, 2, 3, 4, 5\}$ in Amazon). Meanwhile, the review texts are pre-processed to a fixed-length tensor $E \in \mathbb{R}^{M \times N \times d}$, where $\mathbf{e}_{i,j}$ denotes the feature of review text user i comments on item j. Then the user-item interactions $\mathcal{E}$ can be represented by the combination of the rating matrix and the review tensor, i.e., $\mathcal{E} = (R, E)$. Finally, the input data can be formulated as a user-item bipartite graph $G = (\mathcal{U} \cup \mathcal{I}, \mathcal{E})$. The task is to predict the values of the full rating matrix $\hat{R} \in \mathbb{R}^{M \times N}$ based on the graph G.

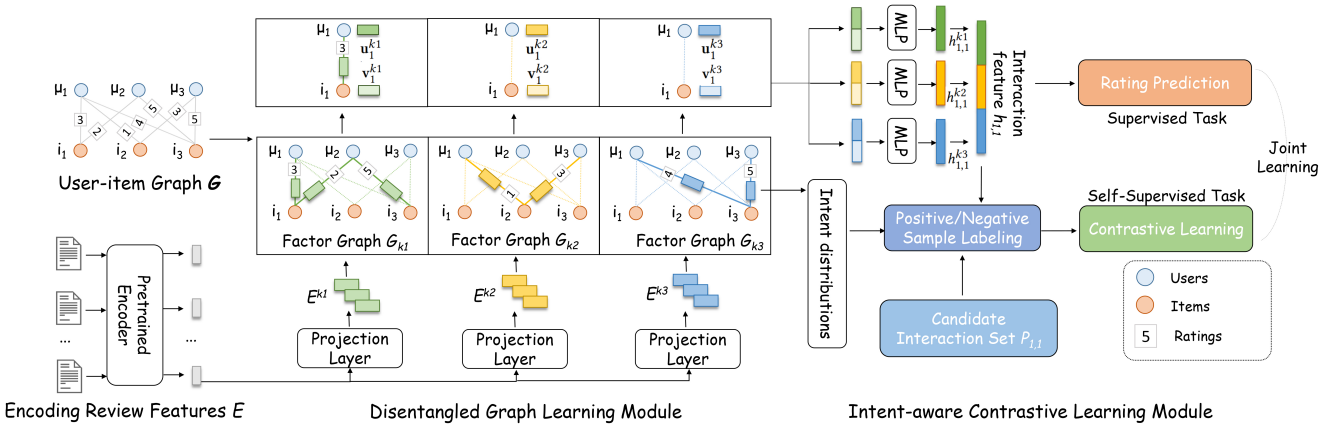


Figure 2: Overview of SGDN. For clarity, we show the pipeline to generate the rating prediction for one user-item interaction.

4 Proposed Model

This section gives a detailed introduction to our proposed SGDN. The overview of SGDN is shown in Figure 2, which is composed of two parts: 1) Disentangled Graph Learning (DGL) Module: factorizing the input graph based on the user-item interactions and learning disentangled representations for users/items and interactions. 2) Intent-aware Contrastive Learning (ICL) Module: introducing an auxiliary CL task to alleviate the sparsity issue and encourage disentanglement. Our demonstration is unfolded as follows.

4.1 Disentangled Graph Learning Module

To model user/item's attributes pertinent to latent factors, we design a GNN model that learns disentangled user/item and interaction representations. Each GNN layer consists of a graph disentangling mechanism which identifies the latent factors in interactions to form multiple factor graphs, and a factorized message passing mechanism that performs multi-channel message passing on the factor graphs. Finally, multiple GNN layers are stacked to gather useful information from higher-order neighborhoods and make rating predictions.

Initialization

We parameterize user/item ID embeddings as free embedding matrices $U \in \mathbb{R}^{M\times d}$ and $V \in \mathbb{R}^{N\times d}$. Then we further divide the ID embedding into K chunks for separate user/item representation learning in each channel. Specifically, the ID embedding for user i is represented as:

$$\mathbf{u}_i^{(0)} = (\mathbf{u}_i^{1,(0)}, \mathbf{u}_i^{2,(0)}, \dots, \mathbf{u}_i^{K,(0)}), \tag{1}$$

where $\mathbf{u}_i^{k,(0)} \in \mathbb{R}^{\frac{d}{K}}$ is the chunked embedding of user i on the k-th latent factor. Analogously, $\mathbf{v}_j^{(0)} = (\mathbf{v}_j^{1,(0)}, \mathbf{v}_j^{2,(0)}, \dots, \mathbf{v}_j^{K,(0)})$ is initialized as the ID embedding for item j.

For review representations, we follow [Shuai *et al.*, 2022] to encode user i's review on item j into the vector $\mathbf{e}_{i,j}$ with BERT-Whitening [Su *et al.*, 2021], whose parameters are frozen during training for time efficiency consideration.

Graph Disentangling

Considering the fruitful semantic information, we propose to mine the distribution of latent factors from review texts. Specifically, given the review embedding $e_{i,j}^k$ of user i to item j in channel k, we present a prototype-based method to obtain the semantic score $\mathbf{se}_{i,j}^k$ that indicates how relevant is the review (i,j) to factor k. We introduce K latent factor prototypes $\{\mathbf{c}_k\}_{k=1}^K$ and the score $\mathbf{se}_{i,j}^k$ is calculated as:

$$\mathbf{se}_{i,j}^k = \frac{\exp(\phi\left(\mathbf{e'}_{i,j}^k, \mathbf{c}_k\right)/\tau)}{\sum_{k'=1}^K \exp(\phi\left(\mathbf{e'}_{i,j}^{k'}, \mathbf{c}_{k'}\right)/\tau)}; \mathbf{e'}_{i,j}^k = \mathbf{e}_{i,j}\mathbf{W}_k, \tag{2}$$

where $W_k \in \mathbb{R}^{d\times d}$ is the transformation parameter matrix, $\phi(\cdot,\cdot)$ denotes the cosine similarity and τ is the temperature hyperparameter. $\mathbf{c} \in \mathbb{R}^{K\times d}$ is learnable parameters and initialized by the K-means clustering on the review features $\mathbf{e}$.

Although the reviews can provide us some hints as to which factor user-item interactions fall into, there might be some missing information. Recalling the example in Figure 1, the review text (d) is general and cannot explicitly reflect the reason for user's rating. To tackle this problem, we propose to infer it from the neighborhood of user/item. Generally, if user i/item j frequently interacts with its neighbors based on factor k, we can draw the inference that user i might also rating item j based on factor k with high probability. On the basis of this insight, we further introduce the similarity between user i and item j in terms of aspect k to assist in judging the latent factors of interactions, which can be formulated as:

$$\mathbf{st}_{i,j}^{k,(l)} = \frac{\exp(\phi\left(\mathbf{u}_i^{k,(l-1)}, \mathbf{v}_j^{k,(l-1)}\right)/\tau)}{\sum_{k'=1}^K \exp(\phi\left(\mathbf{u}_i^{k',(l-1)}, \mathbf{v}_j^{k',(l-1)}\right)/\tau)}, \tag{3}$$

where $\mathbf{st}_{i,j}^{k,(l)}$ denotes the structural score of user i and item j on the k-th factor at the l-th layer and $\mathbf{u}_i^{k,(l-1)}$/$\mathbf{v}_j^{k,(l-1)}$ denotes the learned embedding of user i/item j at the $(l-1)$-th layer in the k-th channel. When $l=1$, $\mathbf{st}^{k,(l)}$ reflects the matching degree of user and item's own attributes on factor k; When $l>1$, $\mathbf{st}^{k,(l)}$ can integrate the information on factor

k from a larger receptive field due to the iterative accumulation of factor-relevant information from neighborhoods via factorized message passing.

Having modeled the distributions of latent factors from both semantic and structural perspectives, we then combine them into the final score $\mathbf{s}_{i,j}^{k,(l)}$ representing the coefficient of the edge between user i and item j in the k-th factor graph:

$$\mathbf{s}_{i,j}^{k,(l)} = \frac{\mathbf{se}_{i,j}^{k}}{1+\exp(-\eta_{i,j})} + \frac{\mathbf{st}_{i,j}^{k,(l)}}{1+\exp(\eta_{i,j})}, \tag{4}$$

where $\eta_{i,j} \in \mathbb{R}$ is a learnable parameter to balance the weights of semantic score and structural score.

Factorized Message Passing

Given the learned factor graphs, we aim to leverage message passing to accumulate factor-relevant information for user/item representation learning. Specifically, we perform embedding propagation [Kipf and Welling, 2016] in each channel, such that the information of reviews and neighboring items/users, which are relevant to the factor, are integrated into the learned user/item representations. Following [Berg *et al.*, 2017], we treat rating score as edge type. Then for rating r, the factorized message passing from item j to user i in the l-th layer is formulated as:

$$\mathbf{x}_{r;j\to i}^{k,(l)} = \frac{\mathbf{s}_{i,j}^{k,(l)}\sigma(\mathbf{e'}_{i,j}^{k}\cdot\mathbf{W}_{r,1}^{k,(l)} + \mathbf{v}_{j}^{k,(l-1)}\cdot\mathbf{W}_{r,2}^{k,(l)})}{\sqrt{\left|\mathcal{D}_{j}^{k,(l)}\right|\left|\mathcal{D}_{i}^{k,(l)}\right|}}, \tag{5}$$

where $W_{r,1}^{k,(l)} \in \mathbb{R}^{d\times\frac{d}{K}}$ and $W_{r,1}^{k,(l)} \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ are the parameter matrices to project the review embedding and user/item embedding to the same space in the k-th channel. $\mathcal{D}_{i}^{k,(l)} = \sum_{p\in\mathcal{N}(i)}\mathbf{s}_{i,p}^{k,(l)}$ and $\mathcal{D}_{j}^{k,(l)} = \sum_{p\in\mathcal{N}(j)}\mathbf{s}_{p,j}^{k,(l)}$ denote the degrees of user i and item j in the l-th layer of channel k.

To intuitively figure out the essence of Equation (5), we hypothesize that factor k represents *price*. Then the interpretation is three-fold: 1) the coefficient $\mathbf{s}_{i,j}^{k,(l)}$ is capable of filtering out the noise information of reviews and items with which user i do not interact due to price. 2) The review information $\mathbf{e}_{i,j}^{k}$ of user i is collected to characterize his/her reviewing behaviors based on price. 3) The neighboring item feature $\mathbf{v}_{j}^{k,(l-1)}$ of user i is accumulated to depict his/her price-sensitive preference on items.

After message passing in each channel, we then aggregate all the factor-relevant messages as follows:

$$\mathbf{u}_{i}^{k,(l)} = \mathbf{W}^{(l)}\sum_{r\in\mathcal{R}}\sum_{p\in\mathcal{N}_{i,r}}\mathbf{x}_{r;p\to i}^{k,(l)}, \tag{6}$$

where $\mathbf{W}^{(l)} \in \mathbb{R}^{\frac{d}{K}\times\frac{d}{K}}$ is the parameter matrix and $\mathcal{N}_{i,r}$ is the set of items that user i rates with rating r. Similarly, we can obtain the aggregated feature $\mathbf{v}_{j}^{k,(l)}$ of item j.

Layer Combination and Prediction

To capture the useful information from higher-order neighbors, we further stack L graph disentangling layers to form the final representations for users/items in each channel:

$$\mathbf{u}_{i}^{k} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{u}_{i}^{k,(l)}; \quad \mathbf{v}_{j}^{k} = \frac{1}{L}\sum_{l=1}^{L}\mathbf{v}_{j}^{k,(l)}. \tag{7}$$

Then we model the interactions between users and items from each latent factor using a Multi-Layer Perceptron (MLP) to obtain the factorized interaction feature $h_{i,j}^{k}$ and merging representations under all factors:

$$\mathbf{h}_{i,j} = \|_{k=1}^{K}\mathbf{h}_{i,j}^{k};\ \mathbf{h}_{i,j}^{k} = \mathrm{MLP}\left(\mathbf{u}_{i}^{k}\|\mathbf{v}_{j}^{k}\right), \tag{8}$$

where $\mathbf{h}_{i,j} \in \mathbb{R}^{d}$ denotes the merged interaction feature between user i and item j. The predicted rating score of user i to item j is calculated as:

$$\hat{r}_{i,j} = \mathbf{w}^{\top}\mathbf{h}_{i,j}, \tag{9}$$

where $\mathbf{w} \in \mathbb{R}^{d}$ is a parameter vector. We employ Mean Square Error (MSE) loss as the supervision signal:

$$\mathcal{L}_{\text{sup}} = \frac{1}{|\mathcal{T}|}\sum_{(i,j)\in\mathcal{T}}\left(\hat{r}_{i,j} - r_{i,j}\right)^{2}, \tag{10}$$

where $\mathcal{T}$ denotes the observed user-item interactions in the training set.

4.2 Intent-aware Contrastive Learning Module

In this section, we design a contrastive learning task to alleviate the sparsity issue of user-item interactions. So far, we have obtained the interaction features coupling with the latent factor distributions $\mathbf{s}_{i,j}^{L} = (\mathbf{s}_{i,j}^{1,L}, \mathbf{s}_{i,j}^{2,L}, \ldots, \mathbf{s}_{i,j}^{K,L})$ from the disentangled graph learning module. A very important step in contrastive learning is to select reasonable positive and negative examples. Previous practice [Shuai *et al.*, 2022] considers the embeddings of the same node/interaction from different views as positive pairs and that of different nodes/interactions as negative pairs. However, we argue that different interactions with the same rating and similar intents should also be treated as positive pairs. Therefore, for each interaction (i,j), we calculate the similarity of latent factor distribution between it and other interactions with the same rating and select the top-k ones as positive samples of (i,j), which is formulated as follows:

$$\begin{aligned}\mathcal{P}_{(i,j)} &= \{(i',j'), r_{i,j} = r_{i',j'}\},\\ \boldsymbol{y}_{i',j'} &= \mathbf{s}_{i,j}^{L\ \top}\mathbf{s}_{i',j'}^{L}, (i',j') \in \mathcal{P}_{(i,j)},\\ \mathcal{P}_{(i,j)+} &= \mathrm{Rank}(\boldsymbol{y}, K_p),\end{aligned} \tag{11}$$

where $\mathcal{P}_{(i,j)}$ denotes the interaction set with the same rating as anchor and $\mathcal{P}_{(i,j)}+$ denotes the set of positive samples. $\mathrm{Rank}(\boldsymbol{y}_{i',j'}, K_p)$ returns the interaction indices of the K_p-largest values in $\boldsymbol{y}$.

As revealed in a recent study [Xia *et al.*, 2022a], CL benefits from hard negatives which are similar to the anchor. In our rating prediction task, interactions with the same rating tend to have similar embeddings under the supervision by the MSE loss. Thus to mine the hard negatives, we select the remaining interactions in $\mathcal{P}_{(i,j)}$ as negative examples. Specifically, we randomly discard edges on graph G with probability p to

Datasets	Toys	Clothing	Office	Kitchen	Tools
#Users	19,412	39,387	4,905	66,519	16,638
#Items	11,924	23,033	2,420	28,237	10,217
#Reviews	167,597	278,677	53,228	551,682	134,476
Density	0.072%	0.031%	0.448%	0.029%	0.079%

Table 1: Statistics of datasets.

generate two different views G_1 and G_2. Then we adopt the InfoNCE loss [Gutmann and Hyvärinen, 2010] to maximize the agreement of positive pairs compared to negative pairs:

$$\mathcal{L}_{ssl} = \sum_{(i,j)\in\mathcal{T}} -\log \frac{\sum_{(i',j')\in\mathcal{P}_{(i,j)}+} \exp(\mathbf{h^1_{i,j}}^T \mathbf{h}^2_{i',j'}/\tau)}{\sum_{(i',j')\in\mathcal{P}_{(i,j)}} \exp\left(\mathbf{h^1_{i,j}}^T \mathbf{h}^2_{i',j'}/\tau\right))},$$

where $h^1_{i,j}$ and $h^2_{i,j}$ are the interaction features from the two views, respectively. By dynamically generating positive and negative pseudo-labels based on the intent similarities, SGDN provides a self-supervised signal to refine the interaction features and further encourage disentanglement.

4.3 Model Optimization

We jointly optimize the recommendation model by combining the above two losses:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + \lambda \mathcal{L}_{ssl}, \tag{12}$$

where λ is a hyperparameter to control the contribution of CL task towards the overall objective.

4.4 Model Complexity Analysis

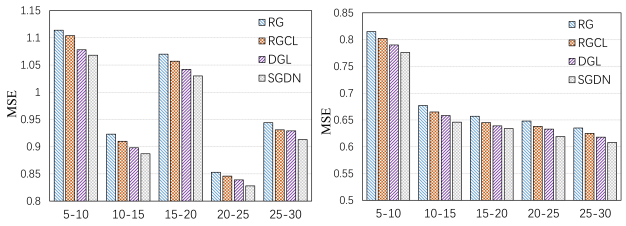
For the memory cost, it is notable that we divide the ID embedding into K chunks to keep that the same as previous works [Liu *et al.*, 2021; Shuai *et al.*, 2022]. The extra parameters involved in the DGL module are $O(|\mathcal{E}| + K \times d \times d)$. For the time cost, the complexity of the DGL module is $O(L \times K \times |\mathcal{E}| \times \frac{d}{K})$ and the complexity of calculating the CL loss in Eq. (12) is $O(B \times |\mathcal{E}| \times d)$. An alternative to reduce the time complexity is selecting positive and negative samples within the batch, reducing the time complexity to $O(B^2 \times d)$. Thus the overall time complexity of SGDN is $O((L \times |\mathcal{E}| + B^2) \times d)$, which is comparable with many GNN-based recommendation methods [Shuai *et al.*, 2022; Xia *et al.*, 2022b].

5 Experiments

5.1 Experimental Settings

Datasets

Following [Shuai *et al.*, 2022], we evaluate SGDN on the Amazon review dataset [He and McAuley, 2016]. *Toys and Games*, *Office Products*, *Clothing*, *Home and Kitchen*, and *Tools and Home Improvement* are the five 5-core subsets selected (shortened as Toys, Office, Clothing, Kitchen, and Tools, respectively). The rating scores for all the five datasets range from 1 to 5. Each dataset is randomly split into training, validation, and test sets with a ratio of 8:1:1. The details of these datasets are summarized in Table 1.



(a) Clothing (b) Office

Figure 3: Performance *w.r.t* interaction degrees.

Baselines

We compare SGDN with the SOTA methods, including CNN-based methods (DeepCoNN [Zheng *et al.*, 2017], TransNet [Catherine and Cohen, 2017] and DRRNN [Xi *et al.*, 2021]), attention-based methods (DAML [Liu *et al.*, 2019] and DRRNN [Xi *et al.*, 2021]), disentanglement-based method (DGCF [Wang *et al.*, 2020]), and graph-based methods (RMG [Wu *et al.*, 2019b], and RGCL [Shuai *et al.*, 2022]).

Parameter Settings

The hyperparameters for the baseline models are tuned according to the original paper. It is notable that we reimplement DGCF by replacing the BPR loss [Rendle *et al.*, 2012] with MSE loss to accommodate the rating prediction task. For SGDN, we use Adam to optimize the parameters with a learning rate of 0.01. The size of embeddings d for users/items and reviews is set as 64. We choose the number of message passing layers L from {1, 2, 3}, the number of latent factors from {2, 4, 8}, and the dropout ratio from {0.7, 0.8, 0.9}. The temperature hyperparameter τ is tuned from {0.2, 0.5, 1}. The hyperparameter λ is searched from {0.01, 0.05, 0.1, 0.5}.

Evaluation Metric

Following [Liu *et al.*, 2019], we evaluate the performance by MSE. Each experiment is repeated five times and the average performance is reported for each dataset.

5.2 Performance Comparison

Overall Performance Comparison

The comparison results of all methods are presented in Table 2 and the following observations can be made:

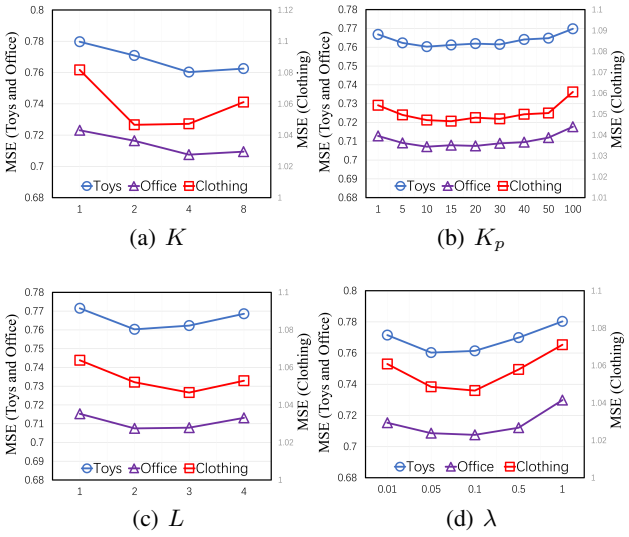
- SGDN achieves the best results on every dataset tested and significantly outperforms the strongest baseline, RGCL, by an average of 2.5%. The improvements of SGDN relative to all other baselines can be attributed to: 1) By disentangling the graph from semantic and structural perspectives, SGDN is able to model user preferences based on multiple latent factors more accurately. 2) The intent-aware CL task can assist SGDN in disentangling the factors and alleviating the problem of the sparsity interactions.
- DGCF achieves comparable or superior performance in comparison to many CNN-based or attention-based baselines despite its ignoring review information, which demonstrates the efficacy of disentangling in review-based recommender systems. Meanwhile, SGDN outperforms DGCF by a large margin, which validates the efficacy of the graph

Datasets	DeepCoNN	TransNet	DRRNN	NARRE	DAML	DGCF	RMG	RGCL	SGDN	Improv.
Toys	0.8026	0.7982	0.7884	0.7961	0.7940	0.7943	0.7901	0.7771	**0.7603***	**2.2%**
Clothing	1.1184	1.1141	1.1035	1.1064	1.1065	1.1002	1.1064	1.0858	**1.0466***	**3.6%**
Office	0.7426	0.7419	0.7306	0.7408	0.7358	0.7345	0.7348	0.7228	**0.7075***	**2.2%**
Kitchen	1.0914	1.0879	1.0769	1.0835	1.0814	1.0798	1.0783	1.0732	**1.0528***	**1.9%**
Tools	0.9356	0.9348	0.9249	0.9304	0.9295	0.9301	0.9288	0.9241	**0.9010***	**2.5%**

Table 2: Comparison results on the five datasets in terms of MSE. The best and second-best results are highlighted with boldface and underlined. * indicates SGDN significantly outperforms the best baseline with $p < 0.05$ using student t-test on the dataset.

Datasets	Toys	Clothing	Office
RG	0.7853	1.1024	0.7293
DGL Variant 1	0.7857	1.0987	0.7287
DGL Variant 2	0.7765	1.0748	0.7193
DGL Variant 3	0.7797	1.0850	0.7246
DGL	0.7721	1.0637	0.7152
DGL+ICL-NP	0.7668	1.0542	0.7128
SGDN	0.7603	1.0466	0.7075

Table 3: Ablation studies on the DGL and ICL modules.



(a) K (b) K_p (c) L (d) λ

Figure 4: Impact of key hyperparameters in SGDN.

disentangling mechanism in SGDN and highlights the importance of review data in disentangling.

Performance Comparison in Alleviating Data Sparsity
To verify the robustness of SGDN against sparsity issue, we partition users into distinct groups according to their interaction numbers in the training set (e.g., 5-10). Then we report the MSE of DGL, SGDN in comparison to the SOTA models RGCL and RG (RGCL minus the CL tasks) for each group. Figure 3 demonstrates that, compared to RG, DGL is more robust to the sparsity issue, allowing for more effective use of review information to disentangle latent factors in user-item interactions. In addition, SGDN improves upon RGCL by selecting proper positive and negative samples based on user's interaction intents in the CL task. Consequently, SGDN achieves the highest performance across all groups, demonstrating the efficacy of our proposed CL task.

5.3 Ablation Studies

In this section, we conduct ablation research on the two modules in SGDN to comprehend their functions more deeply.

Impact of the DGL module. To validate the efficacy of DGL, we compare it with RG, the SOTA graph learning model. In addition to RG, we also compare DGL to its three variants: 1) Variant 1 calculates the coefficient $\mathbf{s}_{i,j}^{k,(l)}$ in a uniform manner, i.e., $\mathbf{s}_{i,j}^{k,(l)} = \frac{1}{K}$. 2) Variant 2 calculates $\mathbf{s}_{i,j}^{k,(l)}$ based on the semantic information, i.e., $\mathbf{s}_{i,j}^{k,(l)} = \mathbf{se}_{i,j}^{k}$. 3) Variant 3 calculates $\mathbf{s}_{i,j}^{k,(l)}$ based on the structural information, i.e., $\mathbf{s}_{i,j}^{k,(l)} = \mathbf{st}_{i,j}^{k,(l)}$. The results are shown on the top of Table 3. The key observations are as follows:

First, Variant 1 almost has no performance improvements compared to RG because it fails to model the intent distributions among interactions. Second, we observe a decrease in the performance of Variant 2 and 3 compared to DGL, demonstrating that integrating semantic and structural information allows for a comprehensive exploration of the distributions of latent factors in interactions. Third, DGL has a significant improvement over RG, validating the efficacy of its disentanglement in user's intents.

Impact of the ICL module. To validate the efficacy of the proposed CL task, we remove the positive and negative sampling strategy in the ICL module (named ICL-NP) and compare it with SGDN. The results are summarized at the bottom of Table 3. We observe that adding self-supervised task can alleviate the sparsity issue and improve the overall effect of DGL. Moreover, SGDN consistently outperforms DGL+ICL-NP, highlighting the efficacy of learned intent distributions in identifying the positive and negative samples.

5.4 Hyperparameter Studies

In this section, we evaluate the effect of key hyperparameters (# of latent factors K, # of positive samples K_p, # of message passing layers L, and importance hyperparameter of CL task λ) in SGDN and show the evaluation results in Figure 4.

- The value of K is examined in $\{1, 2, 4, 8\}$. We find that SGDN performs the worst when $K = 1$, indicating that modeling the user characteristics as a whole is insufficient to capture user behavioral patterns. Increasing the factor number from 1 to 4 significantly enhances the model performance. However, excessive disentanglement leads to

Factor k_1	$r = 1$	$s_{i,j}^{k_1,L} = 0.642$	I feel like this printer has the capabilities of a large office copier printer, but on a slightly smaller scale for home use.
	$r = 3$	$s_{i,j}^{k_1,L} = 0.591$	This product is okay, but i had a difficult time getting it to stay open. i don't think it would be very beneficial in my business.
	$r = 5$	$s_{i,j}^{k_1,L} = 0.580$	It has double pockets to hold papers and this size is good for me in this type of binder.
Factor k_2	$r = 1$	$s_{i,j}^{k_2,L} = 0.638$	When i got this pencil sharpener, the black plastic housing literally fell off the unit and a small white plastic piece and another black plastic gear fell onto the kitchen countertop.
	$r = 3$	$s_{i,j}^{k_2,L} = 0.593$	Print quality seems to be OK, but there 's no way to tell the printer whether you are using plain paper of glossy paper. Plain paper prints look washed out, premium paper prints look good.
	$r = 5$	$s_{i,j}^{k_2,L} = 0.554$	It's solidly made and stands up to regular use pretty darn well. The result is crisp laser printing on a home office budget.
Factor k_3	$r = 1$	$s_{i,j}^{k_3,L} = 0.657$	My screen shows fraud not real hp brands the sent me back one. Others have found same problems and many of the ink tanks don't fit.
	$r = 3$	$s_{i,j}^{k_3,L} = 0.592$	This product almost delivers on its promises one. But the individual packets of labels easily detached from the main package.
	$r = 5$	$s_{i,j}^{k_3,L} = 0.571$	Making photo prints uses a lot of ink. This helps address that problem. Same quality prints as standard capacity cartridge.
Factor k_4	$r = 1$	$s_{i,j}^{k_4,L} = 0.613$	In contrast to the high quality pictures you get from canon machines in the same price range, epson's inks are only fade resistant and they don't tell you how long they'll actually last.
	$r = 3$	$s_{i,j}^{k_4,L} = 0.636$	For a relatively inexpensive laminator, this does an OK job. But the lack of guides on this unit is a real problem.
	$r = 5$	$s_{i,j}^{k_4,L} = 0.539$	I needed to purchase several different cartridges, and the pricing from amazon was favorable, so I made this purchase.

Table 4: Examples of reviews corresponding to each latent factor on *Office*. The key information is highlighted with red.

small performance degradation, which might be attributed to too fine-grained factors and limited expressiveness.

- The value of K_p is tuned in the range from 1 to 100. Figure 4(b) show the results. With the increase of the positive examples, the performance of SGDN almost remains stable on the three datasets when $K_p < 50$. However, too large K_P might lead to the quality drop due to misidentifying the interactions with dissimilar intents as positive samples.
- We investigate the impact of message passing layer number by setting $L \in \{1, 2, 3, 4\}$ and present the results in Figure 4(c). Clearly, SGDN is able to distill useful higher-order information from multi-hop neighbours according to the significant improvements by increasing L from 1 to 2. While continuing the model depth, the improvements are not that obvious or even slightly degrade. This indicates that second-order connectivity might be sufficient to extract factor-relevant information.
- We also study the impact of $\lambda \in \{0.01, 0.05, 0.1, 0.5, 1\}$ that weighs the contribution of CL task. As shown in Figure 4(d), the performances of SGDN first increase and then decrease on the three datasets, following a under- to over-fitting pattern. Generally, SGDN is not very sensitive when λ is tuned in a reasonable range (e.g. 0.05-0.1).

5.5 Explainability Studies

To better interpret the latent semantics of the learned factor graphs, we explain the reasons behind user ratings by presenting the reviews of high-confidence interactions. In particular, we conduct experiments on Office with factor number $K = 4$ and layer number $L = 2$. For each factor k, we randomly select one review with score $\mathbf{s}_{i,j}^{k,L} > 0.5$ for rating 1, 3, 5, respectively. This indicates that factor k dominates user i's rating on item j. We present the reviews and associated scores in Table 4 and have observations as follows:

- By jointly analyzing the reviews of the same latent factor, we find that, despite being written for different items and ratings, they all have inherent semantic connections. For example, the reviews of factor k_1 reflect user ratings based on how well the product meets their needs, such as *good for me* and *beneficial in my business*. The reviews for factor k_2 are all about the quality and durability of the products, such as *quality seems to be OK* and *solidly made*.
- By jointly analyzing the reviews across multiple factors, we find that, SGDN is capable of modeling the interactions from multiple perspectives. In general, we character k_1, k_2, k_3, and k_4 as *demand-supply match*, *quality*, *integrity* and *cost performance*, respectively. This verifies our hypothesis that different uses' ratings on different items are driven by distinct latent factors.

6 Conclusion

This paper proposes a novel framework, SGDN, which focuses on exploring and disentangling the latent factors behind user-item interactions for better and more explainable review-based recommendation. Specifically, we design a disentangled graph learning module to factorize the user-item rating graph and learn disentangled user/item representations. Then an intent-aware contrastive learning task is designed to alleviate the sparsity issue and encourage disentanglement. Experiments on five benchmark datasets validate the superior performance and interpretability of SGDN.

Acknowledgments

This work was supported by NSF China (No. 42050105, 62020106005, 62061146002, 61960206002), Shanghai Pilot Program for Basic Research - Shanghai Jiao Tong University.

References

[Berg *et al.*, 2017] Rianne van den Berg, Thomas N Kipf, and Max Welling. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263*, 2017.

[Catherine and Cohen, 2017] Rose Catherine and William Cohen. Transnets: Learning to transform for recommendation. In *RecSys*, pages 288–296, 2017.

[Chen *et al.*, 2018] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *WWW*, pages 1583–1592, 2018.

[Gutmann and Hyvärinen, 2010] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *ICAIS*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[He and McAuley, 2016] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, pages 507–517, 2016.

[Higgins *et al.*, 2016] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Koren *et al.*, 2022] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pages 91–142, 2022.

[Liu *et al.*, 2019] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *SIGKDD*, pages 344–352, 2019.

[Liu *et al.*, 2021] Yong Liu, Susen Yang, Yinan Zhang, Chunyan Miao, Zaiqing Nie, and Juyong Zhang. Learning hierarchical review graph representations for recommendation. *TKDE*, 2021.

[Ma *et al.*, 2019] Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, and Wenwu Zhu. Disentangled graph convolutional networks. In *ICML*, pages 4212–4221. PMLR, 2019.

[Ma *et al.*, 2020] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. Disentangled self-supervision in sequential recommenders. In *SIGKDD*, pages 483–491, 2020.

[Mao *et al.*, 2016] Mingsong Mao, Jie Lu, Guangquan Zhang, and Jinlong Zhang. Multirelational social recommendations via multigraph ranking. *IEEE transactions on cybernetics*, 47(12):4049–4061, 2016.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[Shuai *et al.*, 2022] Jie Shuai, Kun Zhang, Le Wu, Peijie Sun, Richang Hong, Meng Wang, and Yong Li. A review-aware graph contrastive learning framework for recommendation. *arXiv preprint arXiv:2204.12063*, 2022.

[Su *et al.*, 2021] Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*, 2021.

[Sun *et al.*, 2020] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. Dual learning for explainable recommendation: Towards unifying user preference prediction and review generation. In *WWW*, pages 837–847, 2020.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[Wang *et al.*, 2020] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. Disentangled graph collaborative filtering. In *SIGIR*, pages 1001–1010, 2020.

[Wu *et al.*, 2019a] Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. Hierarchical user and item representation with three-tier attention for recommendation. In *NAACL*, pages 1818–1826, 2019.

[Wu *et al.*, 2019b] Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. Reviews meet graphs: enhancing user and item representations for recommendation with hierarchical attentive graph neural network. In *EMNLP*, pages 4884–4893, 2019.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *SIGIR*, pages 726–735, 2021.

[Xi *et al.*, 2021] Wu-Dong Xi, Ling Huang, Chang-Dong Wang, Yin-Yu Zheng, and Jian-Huang Lai. Deep rating and review neural network for item recommendation. *TNNLS*, 2021.

[Xia *et al.*, 2022a] Jun Xia, Lirong Wu, Ge Wang, Jintao Chen, and Stan Z Li. Progcl: Rethinking hard negative mining in graph contrastive learning. In *ICML*, pages 24332–24346. PMLR, 2022.

[Xia *et al.*, 2022b] Lianghao Xia, Chao Huang, Yong Xu, Jiashu Zhao, Dawei Yin, and Jimmy Huang. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval*, pages 70–79, 2022.

[Zheng *et al.*, 2017] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*, pages 425–434, 2017.