A Review-aware Graph Contrastive Learning Framework for Recommendation

Jie Shuai
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
shuaijie.hfut@gmail.com

Kun Zhang
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
zhang1028kun@gmail.com

Le Wu*
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
Institute of Artificial Intelligence,
Hefei Comprehensive National
Science Center
lewu.ustc@gmail.com

Peijie Sun
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
sun.hfut@gmail.com

Richang Hong
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
Institute of Artificial Intelligence,
Hefei Comprehensive National
Science Center
hongrc.hfut@gmail.com

Meng Wang
Key Laboratory of Knowledge
Engineering with Big Data,
Hefei University of Technology
Institute of Artificial Intelligence,
Hefei Comprehensive National
Science Center
eric.mengwang@gmail.com

Yong Li
Beijing National Research Center for
Information Science and Technology,
Department of Electronic
Engineering, Tsinghua University
liyong07@tsinghua.edu.cn

ABSTRACT

Most modern recommender systems predict users' preferences with two components: user and item embedding learning, followed by the user-item interaction modeling. By utilizing the auxiliary review information accompanied with user ratings, many of the existing review-based recommendation models enriched user/item embedding learning ability with historical reviews or better modeled user-item interactions with the help of available user-item target reviews. Though significant progress has been made, we argue that current solutions for review-based recommendation suffer from two drawbacks. First, as review-based recommendation can be naturally formed as a user-item bipartite graph with edge features from corresponding user-item reviews, how to better exploit this unique graph structure for recommendation? Second, while most current models suffer from limited user behaviors, can we exploit the unique self-supervised signals in the review-aware graph to guide two recommendation components better? To this end, in this paper, we propose a novel Review-aware Graph Contrastive Learning (RGCL) framework for review-based recommendation. Specifically, we first construct a review-aware user-item graph with feature-enhanced edges from reviews, where each edge feature is composed of both the user-item rating and the corresponding review semantics. This graph with feature-enhanced edges can help attentively learn each neighbor node weight for user and item representation learning. After that, we design two additional contrastive learning tasks (i.e., Node Discrimination and Edge Discrimination) to provide self-supervised signals for the two components in recommendation process. Finally, extensive experiments over five benchmark datasets demonstrate the superiority of our proposed RGCL compared to the state-of-the-art baselines.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Recommender systems**.

KEYWORDS

Recommender Systems, Review-based Recommendation, Graph Contrastive Learning

---

Le Wu is the Corresponding author.

1 INTRODUCTION

As a widely adopted recommendation technique, review-based recommendation tries to model users' preferences to items with the consideration of corresponding reviews [6, 47, 54]. Since textual reviews contain more detailed user opinions and item attributes than numerical ratings, they have been demonstrated to be beneficial to mitigate the data sparsity and the cold-start issue [25, 32, 54], thus have drawn much more attention [6, 10, 24] in recommender systems.

Generally speaking, modern recommender systems usually employ two main components to predict users' preferences: user and item embedding learning, followed by the user-item interaction modeling [7, 14, 15, 46]. Compared to users' rating records, the utilization of the auxiliary textual reviews in review-based recommendation can also be categorized into these two classes: *1) Better user/item representation learning by aggregating user (item) reviews, 2) User-item interaction modeling with each user-item review record.* For the former, historical reviews are usually treated as auxiliary description information for user and item embedding learning [6, 10, 24, 54]. Then, the embeddings are fed into a matching network, such as Factorization Machine (FM) [54], to predict target ratings. For example, DeepCoNN [54] concatenates all historical reviews as the descriptive document for user (item). Then, two convolutional neural networks are employed to infer user and item embeddings from their documents. Finally, FM is used to predict final ratings based on the learned user and item embeddings.

However, this historical review usage is too coarse-grained to exploit review information fully. As shown in Figure 1, each review describes the specific preference of the user for one specific item [5] (e.g., review (a) for style and review (b) for price). Therefore, a fine-grained utilization of reviews is developed to incorporate the detailed information at the user-item interaction modeling stage. To distinguish the historical review usage, we name the type of method as *target review* usage since each review is employed to provide a detailed description for corresponding user-item interaction modeling to predict the *target rating*. Unlike historical review usage, target reviews are accessible for observed user-item pairs at the training stage but unavailable at the inference stage. Therefore, researchers proposed how to transfer the available review in the training process to facilitate the user-item interaction learning at the inference stage. For instance, TransNets [5] inserts an MLP into DeepCoNN before FM to learn interaction features and forces the learned interaction features to approximate the target review features in Euclidean space. SDNets [9] replaces Euclidean distance with a discriminator network to align interaction features and target review features. Then, an adversarial min-max game is applied to reach the alignment through the discriminator. Besides, there are other related models exploiting target reviews for better user and item representation learning, such as DualPC [34] and DRRNN [48].

Despite the great progress they have made, we argue that current solutions for review-based recommendation still have some shortcomings. First of all, review-based recommendation naturally forms
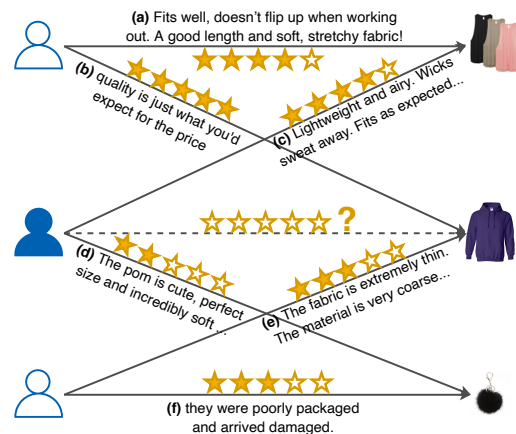
Figure 1: User-item graph with edge features from ratings and reviews.

a user-item bipartite graph with edge features from corresponding user-item reviews. How to explore this unique graph structure for better recommendation is still very challenging. Second, the user/item embedding learning and user-item interaction modeling in recommendations suffer from limited user behaviors, which suppress the recommendation performance. Beyond treating reviews as additional information, is there a better method to develop self-supervised signals (e.g., comparing review representation and user-item interaction representation) for better review utilization in embedding learning and interaction modeling in review-based recommendations? This is another important challenge that we focus on in this paper.

To this end, in this paper, we propose Review-aware Graph Contrastive Learning (RGCL), a novel graph-based contrastive learning framework that fully explores review information for better recommendations. Different from existing graph-based recommendation methods that only employ reviews as supplementation for user/item embedding learning and user-item interaction modeling, we design a novel Review-aware Graph learning (RG) module to integrate reviews into graph learning, where reviews are used to determine the impact of neighbor nodes on the central node in the user-item graph, as well as calculating the impact of reviews to the central node. Along this line, RG can effectively exploit this unique structure of user-item graph with edge features from reviews. Moreover, in order to further exploit this unique structure and make full use of review information in edges, we incorporate Contrastive Learning (CL) to develop self-supervised signals for boosting embedding learning and interaction modeling process based on reviews. Specifically, we design Node Discrimination (ND) and Edge Discrimination (ED) tasks as additional self-supervised optimization targets. ND helps RGCL achieve better node embedding learning by comparing the representations in different sub-graphs. Moreover ED requires RGCL to compare the user-item interaction and corresponding review representation, so that the user-item interaction can be better modeled. Along this line, RGCL is able to fully explore review information for better embedding learning and interaction modeling, which is in favor of item recommendations. Extensive

experiments and detailed analysis over five datasets demonstrate the effectiveness and superiority of our proposed method compared with state-of-the-art review-based baselines. The main contributions of this work can be summarized as follows:

- We define that review-based recommendation can form a user-item graph with edge features from reviews and propose a novel RG to exploit this unique graph structure for better review utilization and graph learning.
- We design a novel RGCL based on contrastive learning, in which ED and ND tasks are employed to use self-supervised signals for the better user (item) embedding learning and user-item interaction modeling based on reviews in edges in the user-item graph.
- We conduct extensive experiments on five real-world datasets to demonstrate the effectiveness of our proposed RGCL.

2 RELATED WORK

2.1 Review-based Recommendation

Given the popular two steps of recommender systems: user/item embedding learning and user-item interaction modeling, review-based recommendation models can also be categorized into *Historical reviews utilization for better embedding learning* and *Target review utilization for better interaction modeling*.

2.1.1 Historical Reviews Utilization. For example, topic models, such as Latent Dirichlet Allocation (LDA) [3], can help researchers obtain latent topics in historical reviews. Thus, topic information can be used to assist the user and item embedding learning. For instance, Collaborative Topic Regression (CTR) [41] used the sum of topic factors and free embeddings to represent items. Topic Initialized latent factors Model (TIM) [28] utilized topic factors to initialize both user and item embeddings. Though these methods have shown the usefulness of textual reviews, LDA technique suffers from insufficient text modeling capacity compared to the modern natural language processing techniques.

To further explore the potential of reviews for better user and item embedding learning, many advanced text methods are introduced into recommendations [10, 31, 35, 47, 54]. For example, Deep Cooperative Neural Networks (DeepCoNN) [54] used two parallel TextCNNs [18] to extract semantic features from user and item historical reviews and achieved impressive performance. To utilize word correlations and review correlations, Paragraph Vector Matrix Factorization (ParVecMF) [1] employed paragraph vector [22] to generate better user and item representations from historical reviews. Moreover, attention mechanism [39, 52] is introduced to improve recommendation performance due to its capability of finding the key elements [53]. For example, Dual attention-based model (D-Attn) is designed to leverage local and global attention layers to select informative words by calculating corresponding attention scores[31]. And Chen et al.[6] proposed NARRE to select relevant reviews in learning the user and item representations with attention mechanism, which can also improve the model interpretability. Besides, the structure of historical reviews also contains useful information. Wu et al.[42] devised a three-tier attention mechanism to make full use of words, sentences, and structure information

in reviews. There are other relevant methods that explore historical reviews for better user and item embedding learning, such as CARL [47], DAML [24], and AHN [10].

Since Graph Convolution Network (GCN) [20, 45] can model the natural user-item bipartite graph and has achieved promising results in item recommendations, researchers also proposed introducing review signals into the learning of graph-based methods. RMG [43] is one of the first few attempts to model user preferences from both review and graph perspectives. It used hierarchical attention networks to extract review features and attention-based GCN to obtain embeddings for nodes (users and items). The prediction is calculated based on the concatenation of learned representations based on review features and node embeddings. Moreover, Gao et al.[11] designed a novel Set-Sequence-Graph (SSG) network to jointly model the multi-view historical reviews, user-item sequences, and the user-item bipartite graph for better user and item embedding learning.

We also borrow the natural user-item graph structure for review-based recommendation, and we advance these related works by introducing review information and constructing a user-item graph with edge features from corresponding reviews for better user preference modeling and item recommendation.

2.1.2 Target Reviews Utilization. Apart from using historical reviews for better embedding learning, incorporating reviews into the user-item interaction modeling stage also attracts much attention. Since it leverages review information to boost the corresponding user-item interaction model in a detailed manner, we name this type of method as target review utilization. Different from historical review utilization, target reviews are only accessible for observed user-item pairs at the training stage but unavailable at the inference stage. To overcome this problem, Catherine and Cohen[5] designed a novel TransNet to make full use of learned information to approximate the target review so as to make a better recommendation. Specifically, they proposed a transform layer to approximate target review features based on learned user-item interactions at the training stage. Euclidean distance is used to constrain the similarity between target reviews and user-item interaction vectors. Some relevant works are also proposed following this approximation strategy, such as DualPC [34] and DRRNN [48]. Besides, some researchers proposed generation-based methods to tackle the problem of unavailable target reviews at the inference stage. For example, SDNet [9] focused on the consistency of target reviews and user-item interactions. It employed GAN to estimate the divergence between target reviews and learned user-item interaction representations so that the target review can be better generated for final prediction.

Despite the great progress they have achieved, current review-based methods only treat reviews as additional information and still suffer from limited user behaviors. Different from these methods, we design two CL tasks to develop self-supervised signals to alleviate the limited user behavior problem and achieve better user preference modeling as well as item recommendation.

2.2 Contrastive Learning

As one of the representative technologies for self-supervised learning, Contrastive Learning (CL) has made great progress in Computer

Vision [16, 38] and Natural Language Processing [1, 26]. The central intuition of contrastive learning is to pull together an anchor and a "positive" sample in embedding space and push apart the anchor from many "negative" samples [2]. In recent years, this idea has attracted researchers to explore CL for graph representation learning [13, 40, 50, 55]. For instance, Veličković et al.[40] proposed Deep Graph Infomax (DGI) by maximizing the agreement between node representation and graph representation in a local-global contrastive paradigm. Zhu et al.[55] developed GRACE to generate node representations by maximizing node-level agreement from edge removing and node feature masking augmentations. You et al.[50] designed GraphCL to explore different types of graph augmentations (such as nodes dropping and edges removing) and analyzed the impacts of various augmentation combinations.

When it comes to graph-based recommendations, researchers designed various CL paradigms at the user-item interaction modeling stage for performance improvement. For example, Cao et al.[4] considered community structures and designed BiGI to recognize the global properties of the bipartite graph. Wu et al.[44] unified graph CL tasks and the recommendation task to enhance user and item representation learning. There also exist other related works [49, 51]. They all have achieved impressive performance on graph-based recommendations, which demonstrate the superiority of CL in graph representation learning. While most of these works focus on designing graph-based self-supervised learning in collaborative filtering, we utilize the unique self-supervised signals in the review-aware graph for recommendation.

3 PROBLEM DEFINITION

In a review-based recommendation system, there are two sets of entities: a user set $\mathcal{U}(|\mathcal{U}| = M)$ and an item set $\mathcal{V}(|\mathcal{V}| = N)$, where $u_i \in \mathcal{U}$ and $v_j \in \mathcal{V}$ denote the user i and item j. M and N denote the number of users and items separately. The rating record is represented as a user-item rating matrix $R \in \mathcal{R}^{M \times N}$, where each element r_{ij} represents the rating score of user i to item j, and $\mathcal{R}$ is the set of all ratings (e.g., $\mathcal{R} = \{1, 2, 3, 4, 5\}$ in Amazon). Meanwhile, we leverage a pre-trained model to process the review that user i commented to item j and obtain a fixed-size vector $e_{ij} \in \mathbb{R}^d$ as the representation. Then, review features of all interactions could be formulated as a tensor $E \in \mathbb{R}^{M \times N \times d}$. The rating matrix R and review tensor E make up the user-item interactions $\mathcal{E} = \{R, E\}$. To this end, the review-based recommendation data can be formulated as a bipartite graph with featured edges $\mathcal{G} = < \mathcal{U} \cup \mathcal{V}, \mathcal{E} >$, with each observed edge containing both the detailed rating value and the semantic vector obtained from the corresponding review. The task of review-based recommendation is to predict the final rating matrix $\hat{R} \in \mathcal{R}^{M \times N}$ with the graph $\mathcal{G}$.

4 REVIEW-AWARE GRAPH CONTRASTIVE LEARNING

Figure 2 illustrates the overall architecture of our proposed RGCL, which consists of two main modules: 1) *Review-aware Graph Learning Module*: incorporating the reviews into user preference modeling based on graph input; 2) *Graph Contrastive Learning Module*: introducing CL technology to promote recommendation performance from two perspectives: user/item embedding learning and

interaction modeling. In the following, we will give a detailed explanation of the two modules.

4.1 Review-aware Graph Learning Module

As mentioned before, we aim at fully exploiting review information to learn accurate user preferences. Specifically, we design a review-aware graph learning module consisting of two key components in recommendation systems: embedding learning and interaction modeling. The module has two advances as follows. Firstly, it naturally formulates the available data as a user-item bipartite graph with ratings and reviews. Besides, we construct a review-aware graph learning that could better capture the correlation between collaborative signals and reviews. Next, we introduce each component in detail.

4.1.1 Review-aware User and Item Embedding Learning. There are three components: *Initialization*: initializing all the inputs, including node embeddings and edge features; *Review-aware Message Passing*: leveraging rating and review information on edges to measure the influence of neighbor nodes and historical reviews to the central node; *Message Aggregation*: aggregating messages from neighbors for the central node embedding learning. The details of each component are listed in the following parts.

Initialization. In this component, we leverage free embedding matrices $U \in \mathbb{R}^{M \times d}$ and $V \in \mathbb{R}^{N \times d}$ to denote users and items. $u_i \in \mathbb{R}^d$ and $v_j \in \mathbb{R}^d$ denote free embeddings of the user i and item j, specifically. Following [37], we treat rating score as edge type to distinguish the semantics carried by ratings. For review representations, similar to [17], we employ BERT-Whitening [33] to generate the representation $e_{ij} \in \mathbb{R}^d$ for the review that user i commented on item j [1]. We have to note that the review representation will be frozen during the model training to reduce the training time and memory usage.

Review-aware Message Passing. As mentioned above, each interaction contains a rating and a review semantic feature. Compared to the numerical rating, textual reviews contain fine-grained semantics, which has following advantages. First, the review semantics contain detailed user preferences and item attributes that can help learn better user and item embeddings [6, 54]. Second, the detailed review semantics can help the model learn the precise extent users like/dislike the items, which can be further employed to re-weight the impacts between users and items. Thus, as shown in Figure 2, we utilize review features to fine-tune the influences of neighbor j and review itself on the central node i, which can be formulated as the rating-specific message passing:

$$x_{r:j \to i}^{(l)} = \frac{\sigma(w_{r,1}^{(l)\top} e_{ij}) W_{r,1}^{(l)} e_{ij} + \sigma(w_{r,2}^{(l)\top} e_{ij}) W_{r,2}^{(l)} v_j^{(l-1)}}{\sqrt{|\mathcal{N}_j||\mathcal{N}_i|}}, \quad (1)$$

where e_{ij} is the review representation that user i commented on item j. $v_j^{(l-1)}$ denotes the embedding of item j learned from the $l-1$ layer, where $v_j^{(0)}$ is initialized with free embedding v_j. $\{w_{r,1}^l, w_{r,2}^l, W_{r,1}^l, W_{r,2}^l \mid r \in \mathcal{R}\}$ are rating-specific trainable parameters at the l-th propagation layer based on rating r. Among them,

---

[1]Compared with concatenation of raw review text, pre-encoded reviews that embody the corresponding ratings have been proved to reduce the training time and the memory usage without decreasing recommendation performance [17].
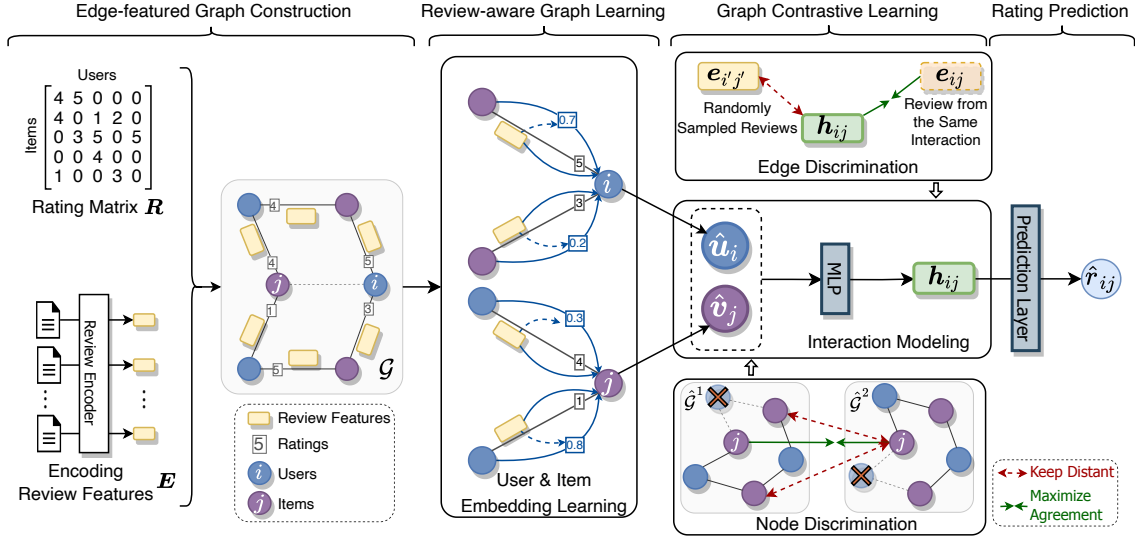
Figure 2: The architecture of Review-aware Graph Contrastive Learning framework.

$W^l_{r,1}$ and $W^l_{r,2}$ are used to transform the review representation e_{ij} and neighbor node embedding v_j to the same space. $w^l_{r,1}$ and $w^l_{r,2}$ are used to learn two scalar weights from the review feature e_{ij} to re-weight the impacts of the neighbor node and review itself on the central node. $\sigma(\cdot)$ is the sigmoid activation function. $\mathcal{N}_i$ and $\mathcal{N}_j$ represent the set of items rated by user i and the set of users who rated item j. Meanwhile, we can obtain the message passing from user i to item j in an analogous way:

$$x^{(l)}_{r;i \to j} = \frac{\sigma(w^{(l)\top}_{r,1} e_{ij}) W^{(l)}_r e_{ij} + \sigma(w^{(l)\top}_{r,2} e_{ij}) W^{(l)}_{r,2} u^{(l-1)}_i}{\sqrt{|\mathcal{N}_i||\mathcal{N}_j|}}, \quad (2)$$

Message Aggregation. Similar to the aggregation operation in GC-MC, we use the following operation to aggregate all messages together to generate user and item embeddings at the l-th layer after the message passing operation:

$$u^{(l)}_i = W^{(l)} \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{N}_{i,r}} x^{(l)}_{r;k \to i}, \quad v^{(l)}_j = W^{(l)} \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{N}_{j,r}} x^{(l)}_{r;k \to j}, \quad (3)$$

where $u^{(l)}_i$ and $v^{(l)}_j$ are the learned embeddings at the l-th layer for user i and item j. $W^{(l)}$ is the trainable parameter matrix. $\mathcal{N}_{i,r}$ denotes the set of items that rated by user i with rating r. $\mathcal{N}_{j,r}$ is the set of users that rated item j with rating r.

By stacking L layers of message passing and aggregation, we take the outputs $u^{(L)}_i$ and $u^{(L)}_j$ from the L-th layer as the final user and item embeddings:

$$\hat{u}_i = u^{(L)}_i, \quad \hat{v}_j = v^{(L)}_j, \quad (4)$$

4.1.2 Interaction Modeling. To model the interaction from user and item embeddings, we take the outputs $\hat{u}_i$ and $\hat{v}_j$ as the input, and leverage an Multi-Layer Perceptron (MLP) to obtain corresponding

interaction features h_{ij}, which can be formulated as follows:

$$h_{ij} = \text{MLP}\left(\left[\hat{u}_i, \hat{v}_j\right]\right), \quad (5)$$

where $h_{ij} \in \mathbb{R}^d$ is the learned user-item interaction features. $\text{MLP}(\cdot)$ denotes the MLP with two hidden layers and GELU activation function. $[,]$ denotes the concatenation operation. After getting h_{ij}, we can predict the rating based on the interaction features:

$$\hat{r}_{ij} = w^\top h_{ij}, \quad (6)$$

where w is a parameter vector that map interaction features to the prediction ratings.

4.2 Contrastive Learning Module

In the above parts, our proposed RG takes both advantages of textual reviews and graph signals for user and item representations. Following the supervised learning framework [6, 37], we can optimize parameters in RG by encouraging the predicted ratings to be close to the observed ratings. However, the limited interaction behaviors restricted the accurate user preference learning, leading to recommender models being far from satisfactory. To alleviate this shortcoming, we intend to leverage Contrastive Learning (CL) to develop self-supervised signals to boost RG learning. Specifically, we design two CL paradigms, i.e., *Node Discrimination* (ND) to enhance node embedding learning and *Edge Discrimination* (ED) to boost interaction modeling.

4.2.1 Node-based CL for Node Representation Enhancement. Following previous work that applies the CL to graph learning [13, 50, 55], we also utilize the CL paradigm to promote better learning of graph structure and the node (i.e., users and items) embedding learning. Specifically, we employ node dropping to generate sub-graphs for contrastive learning. Taking the user node embedding learning as an example, we randomly discard item nodes and corresponding review features with probability ρ to generate sub-graphs $\hat{\mathcal{G}}^1$ and $\hat{\mathcal{G}}^2$ for message passing and aggregation. By applying our

proposed RG to these two sub-graphs, we can obtain the two different embeddings $\hat{u}_i^1$ and $\hat{u}_i^2$ for user i, which can be treated as the positive pair. Similarly, we can obtain the negative node embedding $\hat{u}_{i'}^2$ of different user i' from sub-graph $\hat{\mathcal{G}}^2$, where $i' \neq i$. Therefore, the target is maximizing the similarity of the positive pair $(\hat{u}_i^1, \hat{u}_i^2)$ and minimizing the similarity of the negative pair $(\hat{u}_i^1, \hat{u}_{i'}^2)$. This process can be formulated as follows:

$$\mathcal{L}_{nd}^{user} = -\mathbb{E}_{\mathcal{U}}\left[\log\left(F\left(\hat{u}_i^1, \hat{u}_i^2\right)\right)\right] + \mathbb{E}_{\mathcal{U}\times\mathcal{U}'}\left[\log\left(F\left(\hat{u}_i^1, \hat{u}_{i'}^2\right)\right)\right], \quad (7)$$

where i is the input user, i' is the negative user sampled from $\mathcal{U}' = \mathcal{U}$. And $F\left(\hat{u}_i^1, \hat{u}_i^2\right) = \sigma\left(\hat{u}_i^{1\top} W \hat{u}_i^2\right)$ is the similarity function with trainable parameter W and sigmoid activation function σ. Analogously, we can obtain the optimization target on item nodes $\mathcal{L}_{nd}^{item}$. Combining the ND tasks on both user and item nodes, we get the final ND target: $\mathcal{L}_{nd} = \mathcal{L}_{nd}^{user} + \mathcal{L}_{nd}^{item}$.

4.2.2 Edge-based CL for Target Review Alignment. The review on each edge carries a more detailed assessment from the user to the item than the numerical ratings. To exploit the effective reviews for better interaction modeling, we propose to leverage Edge Discrimination (ED) as the additional CL task to achieve this goal.

After obtaining the interaction feature h_{ij} from Equation 5, we treat it as the anchor example. Following the CL framework [8], we select the corresponding review representation e_{ij} as the positive sample. Naturally, a randomly sampled review representation $e_{i'j'}$ from the entire training set is treated as the negative sample. We have to note that the conditional sampled method, such as selecting the other reviews that user i commented on or item j received, makes no difference with the randomly sampling. The target of ED is to pull together anchor example h_{ij} and positive example e_{ij}, as well as push away anchor example h_{ij} and negative example $e_{i'j'}$. Thus, the optimization target can be formulated as follows:

$$\mathcal{L}_{ed} = -\mathbb{E}_{\mathcal{E}}\left[\log\left(F\left(h_{ij}, e_{ij}\right)\right)\right] + \mathbb{E}_{\mathcal{E}\times\mathcal{E}'}\left[\log\left(F\left(h_{ij}, e_{i'j'}\right)\right)\right], \quad (8)$$

where e_{ij} is the target review feature corresponding to the interaction (i, j). And $e_{i'j'}$ is the negative review feature corresponding to the interaction (i', j') sampled from $\mathcal{E}' = \mathcal{E}$.

4.3 Model Optimization

Since RGCL focuses on predicting the ratings of users to items, we employ Mean Square Error (MSE) as the optimization target, which is widely adopted in current rating prediction works [29, 54]:

$$\mathcal{L}_{main} = \frac{1}{|\mathcal{S}|} \sum_{(i,j)\in\mathcal{S}} (\hat{r}_{ij} - r_{ij})^2, \quad (9)$$

where $\mathcal{S}$ represents user-item pairs in the training set. r_{ij} is the observed rating that user i commented on item j. Meanwhile, we employ CL to constrain the learning process. We optimize the recommendation and CL tasks simultaneously with the importance hyper-parameters α and β:

$$\mathcal{L} = \mathcal{L}_{main} + \alpha\mathcal{L}_{ed} + \beta\mathcal{L}_{nd}. \quad (10)$$

Table 1: Statistics of datasets.

Datasets	#Users	#Items	#Reviews	Density
Digital_Music	5,541	3,568	64,706	0.330%
Toys_and_Games	19,412	11,924	167,597	0.072%
Clothing	39,387	23,033	278,677	0.031%
CDs_and_Vinly	75,258	64,443	1,097,592	0.023%
Yelp	8,423	3,742	88,647	0.281%

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 Datasets. We evaluate our model on five benchmark datasets. Four of them belong to Amazon 5-core [27][2] in different domains: "Digital Music", "Toys and Games", "Clothing", and "CDs and Vinyl". The 5-core means there are at least five reviews for each user or item. Following [31], we also conduct experiments on Yelp Business Rating Prediction Challenge 2013 dataset[3], which contains restaurant reviews in Phoenix, AZ metropolitan area. We have preprocessed the raw data of Yelp following the 5-core setting. The rating scores of these five datasets are integers from one to five. Following similar works [5, 24], each dataset is randomly split into training, validation, and test sets with 80%, 10%, 10%, respectively. The statistics of these five datasets are summarized in Table 1.

5.1.2 Evaluation Metric. Following [9, 47], the performance is evaluated by *MSE*, which is widely used for rating prediction in recommendation systems. For fair comparison, we repeat each experiment **five times** and report *mean(±std)* for model comparison. For review-based recommendation, a relative improvement above 1% is regarded as significant [23, 36].

5.1.3 Baselines. We select conventionally and recently published review-based baselines for model comparison, including advanced graph-based methods. They are listed as follows:

- **SVD** [21] is a classical matrix factorization model that estimates ratings by the inner product of users' and items' latent factors.
- **NCF** [15] uses a neural network to predict the rating based on user and item free embeddings.
- **DeepCoNN** [54] is one of the pioneer works that extracts user/item feature from documents (concatenations of reviews) using neural networks.
- **NARRE** [6] improves upon DeepCoNN by hiring an attention mechanism to estimate the usefulness of different reviews.
- **DAML** [24] enhances user and item representation by modeling the interaction between user and item documents.
- **SDNet** [9] proposes a GAN-based [12] distillation method to transform informative target review signal into NCF.
- **TransNets** [5]: inserts an MLP into DeepCoNN to transform user and item features to an approximation of target review features.

Table 2: Results in terms of the MSE on five datasets of different methods.

Method	Digital_Music	Toys_and_Games	Clothing	CDs_and_Vinly	Yelp
(1) SVD	0.8523±4e-4	0.8086±1e-3	1.1167±1e-3	0.8662±2e-4	1.1939±1e-3
(2) NCF	0.8403±5e-3	0.8078±2e-3	1.1094±1e-3	0.8781±1e-3	1.1896±4e-3
(3) DeepCoNN	0.8378±1e-3	0.8028±7e-4	1.1184±2e-3	0.8621±1e-3	1.1877±1e-3
(4) NARRE	0.8172±1e-3	0.7962±1e-3	1.1064±1e-3	0.8495±1e-3	1.1862±1e-3
(5) DAML	0.8237±2e-3	0.7936±4e-3	1.1065±2e-3	0.8483±1e-3	1.1793±1e-3
(6) SDNet	0.8331±3e-3	0.8006±1e-3	1.1080±1e-3	0.8654±5e-4	1.1837±3e-3
(7) TransNets	0.8273±5e-3	0.7980±1e-2	1.1141±5e-3	0.8440±1e-3	1.1855±2e-3
(8) GC-MC	0.8090±1e-3	0.7986±5e-4	1.1088±1e-3	<u>0.8404</u>±1e-3	1.1737±1e-3
(9) RMG	<u>0.8074</u>±1e-3	<u>0.7901</u>±1e-3	<u>1.1064</u>±2e-3	0.8425±8e-4	<u>1.1705</u>±1e-3
(10) SSG	0.8218±2e-3	0.8064±1e-3	1.1228±1e-3	0.8458±1e-3	1.1807±1e-3
(11) RG	0.8037±2e-3 (0.5%)	0.7853±8e-4 (0.6%)	1.1024±9e-4 (0.4%)	0.8360±1e-3 (0.5%)	1.1692±2e-3 (0.1%)
(12) RG+ND	0.7780±2e-3 (3.6%)	0.7831±1e-3 (0.9%)	1.0925±3e-4 (1.3%)	0.8240±6e-4 (2.0%)	1.1625±1e-3 (0.7%)
(13) RG+ED	0.7810±3e-3 (3.3%)	0.7797±9e-4 (1.3%)	1.0891±8e-4 (1.6%)	0.8244±1e-3 (1.9%)	1.1636±1e-3 (0.6%)
(14) RGCL	**0.7735±4e-3 (4.2%)**	**0.7771±1e-4 (1.6%)**	**1.0858±1e-3 (1.9%)**	**0.8180±7e-4 (2.7%)**	**1.1609±8e-4 (0.8%)**

The best results are highlighted in bold. The percentages indicate the relative improvements over the best baselines marked by underline. All the results are reported as "mean(±std)" across 5 random runs.

- **GC-MC** [37] regards rating prediction as link prediction on the user-item bipartite graph and adopts relational graph convolution [30] to encode user and item embeddings.
- **RMG** [43] is one of the first models that fuse graph signals and review information.
- **SSG** [11] jointly models review sets, review sequences, and user-item graphs. The authors design the Review-aware graph attention network (RGAT) to capture the graph signals for the user-item graph.

Above mentioned methods can briefly categorized into five groups: (1) *Traditional rating-based collaborative filtering methods*, SVD [21] and NCF [15]; (2) *Historical review-based methods*: DeepCoNN [54], NARRE [6], DAML [24]; (3) *Target review-based methods*: TransNets [5], SDNet [9]; (4) *Graph-based method*: GC-MC [37]. (5) *Graph and review fusing methods*: RMG [43] and SSG [11]. We have to note that we reimplement RMG, replace the attentive graph neural with GC-MC, and assign separate convolution channels for each rating of RGAT in SSG for a fair comparison.

5.1.4 Implementation Details. For reviews, we leverage BERT-Whitening [33] to encode each review to a fixed-size feature vector, which will not be updated during model training. In the final architecture of RG , we utilize one layer message passing. The size of embeddings (users, items, and reviews) is set as $d = 64$. We have tested the hyper-parameters α in a range of {0.2, 0.4, 0.6, 0.8, 1.0, 2.0}, β in a range of {0.2, 0.4, 0.6, 0.8, 1.0} and node dropout ration in {0.6, 0.7, 0.8, 0.9}. All the trainable parameters are initialized with the Xavier method, and we select Adam [19] as the optimizer for the entire model training. The entire model is implemented with Deep Graph Library[4] and Pytorch[5] based on Nvidia RTX GPU.

5.2 Performance Evaluation

*5.2.1 Overall Performance Comparison.* Table 2 reports the overall results among five datasets. According to the results, we can obtain the following observations:

Firstly, review-based baselines (Table 2 (3)-(7)) achieve impressive performance, proving the effectiveness of reviews. Moreover, target reviews can improve model performance due to their high relevance with users and items.

Secondly, graph-based baselines (Table 2 (8)-(10)) are able to model complex user-item interactions, thus achieving the best performance over all baselines. This phenomenon supports the importance of graph learning in dealing with higher-order signals. Moreover, we can obtain that the performance of SSG is not as good as the other two graph-based baselines. We speculate the possible reason is that SSG ignores the importance of collaborative filter signals among users and items.

Thirdly, Our proposed RGCL achieves the best performance across all datasets. Compared with all baselines, RGCL has two advantages to achieve the best performance. First of all, RGCL utilizes a newly designed RG to integrate the advantages of historical reviews in graph learning, so that the relevant reviews can be fully explored and the noise problem in reviews will be alleviated. Moreover, we propose to employ two CL tasks (i.e., ND and ED) for better node embedding and interaction feature learning. These two additional CL tasks can help RG to pay close attention to target review signals and embedding learning for user and item, which is in favor of user preference modeling and final rating prediction.

5.2.2 Performance under Different Data Sparsity. As mentioned in Section 1, review information can be used to alleviate the data sparsity problem in recommendation. The overall experimental results in Table 2 have proven that RGCL can make full use of reviews to alleviate this problem and provide better recommendations. In order to further analyze how RGCL achieves this goal, we have conducted additional experiments with different sparsity settings. For

---

[4]https://www.dgl.ai
[5]https://pytorch.org
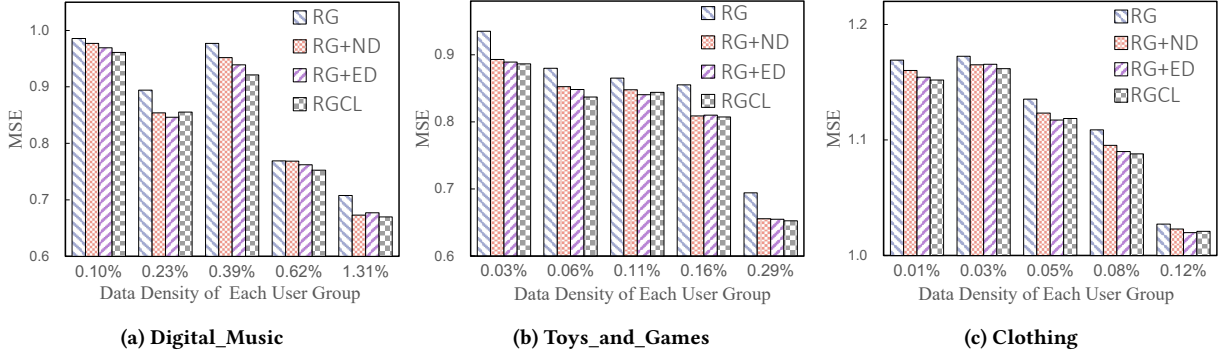
Figure 3: Performance comparison over different user groups among RG, RG+ND, RG+ED and RGCL. The percentages on the abscissa represent the density in the group.

users in training the set, we split them into five groups according to their interaction numbers with items. Specifically, we rank the users on each dataset according to their interaction numbers, and then we uniformly split all users into five groups. Then, we report the MSE comparison of RG, RG+ND, RG+ED, and RGCL in these user groups, as illustrated in Figure 3. According to these results, we have following observations:

Firstly, CL tasks can make better use of reviews and alleviate the data sparsity problem to some extent. Moreover, compared with ND, ED focuses on the consistency of user-item interaction and corresponding review representation, which can achieve better utilization of review information.

Secondly, Our proposed RGCL leverages RG to integrate the advantages of historical review and target review utilization, as well as employ two CL tasks for better users/items embedding and interaction modeling. Therefore, RGCL is able to effectively alleviate the data sparsity problem and have the best performance across different data sparsity settings.

5.3 The Effectiveness of RG

We design a novel RG in RGCL to better integrate the advantages of historical review utilization and target review utilization. To better investigate the effectiveness of RG, we conducted an ablation study on different operations (i.e., weighted review features and the re-weight operation) in RG. The results are illustrated in Table 3, where *RG (w/o review)* denotes that reviews are only used to calculate the weight of corresponding neighbors for the central node, and *RG (w/o weight)* denotes that reviews are only used as impact factors of the central node. We list the detailed message passing of *RG (w/o review)* and *RG (w/o weight)* as follows:

$$RG(w/o\ review): x_{r;j\to i}^{(l)} = \frac{\sigma(w_{r,2}^{(l)\top} e_{ij}) W_{r,2}^{(l)} v_j^{(l-1)}}{\sqrt{|\mathcal{N}_j||\mathcal{N}_i|}},$$

$$RG(w/o\ weight): x_{r;j\to i}^{(l)} = \frac{\sigma(w_{r,1}^{(l)\top} e_{ij}) W_{r,1}^{(l)} e_{ij} + W_{r,2}^{(l)} v_j^{(l-1)}}{\sqrt{|\mathcal{N}_j||\mathcal{N}_i|}}.$$

From the results, we can observe that both *RG(w/o review)* and *RG(w/o weight)* have better performance than GC-MC and RGAT,

Table 3: MSE comparison of different components of RG

Models	Digital_Music	Toys_and_Games	Clothing
GC-MC	0.8145±1e-3	0.8034±5e-4	1.1088±1e-3
RGAT	0.8248±2e-3	0.8093±3e-3	1.1158±2e-3
RG (w/o review)	0.8077±1e-3	0.7927±1e-3	1.1038±1e-3
RG (w/o weight)	0.8074±2e-3	0.7901±1e-3	1.1064±1e-3
RG	0.8037±2e-3	0.7853±8e-4	1.1024±9e-4

indicating that our proposed RG can better explore review information. Moreover, *RG(w/o review)* has better performance than *RG(w/o weight)*. It demonstrates that reviews are critical for determining the weights of impacts from different neighbor nodes to the central node. The results of RG demonstrate that integrating both operations on reviews can fully explore the advantages of historical review utilization, thus improving the model performance effectively.

Furthermore, we conduct experiments to investigate the impact of the message passing layer number L. In detail, we set the number of layers in 1, 2, 3 and take the user and item embeddings at the last layer as the final embeddings. Then, we report their MSE results in Table 4. The MSE results with two layers have a slightly decrease than the performance with one layer. And MSE with three layers shows a relative lager performance reduction. RG achieves the smallest MSE with 1 layer propagation. We speculate the possible reason is that the review feature propagates multi-layers may bring more irrelevant information to node representations. Apart from simply treating the output of the L-th layer as the final representations of users and items, inspired by LR-GCCF [7], we also concatenate the output representations of all L layers of the users and items, then we treat the concatenated representations as users' and items' final representations. And we also find our proposed model only achieves the best performance when L is set to 1.
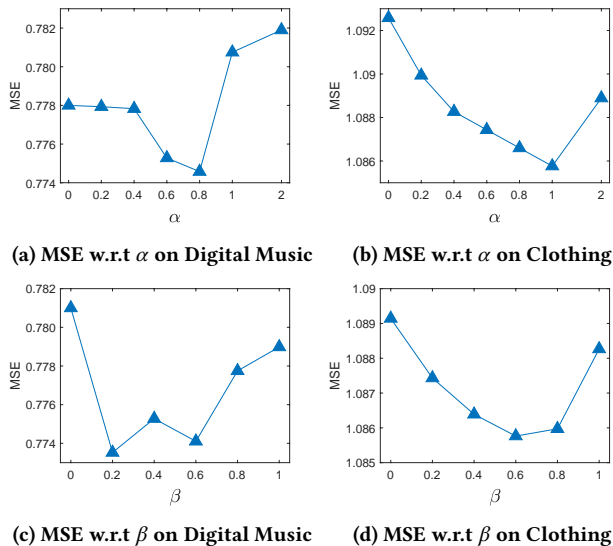
5.4 The Effectiveness of CL tasks

Apart from RG, we also employ two CL tasks (i.e., ED and ND) to help RGCL learn better user and item embeddings. Thus, we

Table 4: MSE comparison of RG with different layers

#Layers	Digital_Music	Toys_and_Games	Clothing
1 Layer	0.8037±2e-3	0.7853±8e-4	1.1024±9e-4
2 Layers	0.8057±1e-3	0.7894±2e-3	1.1067±3e-3
3 Layers	0.8123±1e-3	0.7942±3e-3	1.1144±3e-3

(a) MSE w.r.t α on Digital Music (b) MSE w.r.t α on Clothing

(c) MSE w.r.t β on Digital Music (d) MSE w.r.t β on Clothing

Figure 4: Rating prediction performance with different hyper-parameters on Digital Music and Clothing datasets.

intend to verify the influence of two CL tasks on overall performance. Specifically, we conduct two different types of experiments: 1) *Parameter Sensitive Experiments*: investigating the impact of two CL tasks on the performance of RGCL; 2) *Generalization of CL Tasks*: verifying the generalization of these two CL tasks on other review-based methods. 3) *Quantitative Analysis about the Learned Interaction Features*: conducting quantitative analysis to evaluate the learned interaction features based on these two CL tasks.

5.4.1 Parameter Sensitive Experiments. To investigate the impact of ED and ND on the performance of RGCL, we conduct parameter sensitive experiments for two hyper-parameters α and β on Digital_Music and Clothing datasets. The value of α is in the range {0, 0.2, 0.4, 0.6, 0.8, 1.0, 2.0} and β is in the range {0, 0.2, 0.4, 0.6, 0.8, 1.0}. Corresponding results are summarized in Figure 4.

For the strength of ED loss (i.e., α), we can obtain that model performance first increasing and then decreasing. The best values of α on these two datasets are 0.8 and 1.0, which is consistent with our assumption. ED can help the model focus on the consistent between user-item interaction and learned corresponding review embedding, which is very important for integrating review information for final prediction. Moreover, the target of RGCL is to utilize learnt user and item embeddings to predict the final rating. If the value of α is too large, it will confuse the optimizing process of RGCL, which in turn limits the model performance.

Table 5: MSE of our CL constraints on different recommendation models.

Models	Digital_Music	Toys_and_Games	Clothing
NARRE	0.8172±1e-3	0.7962±1e-3	1.1064±1e-3
+ED	0.8018±4e-3 (1.9%)	0.7910±1e-3 (0.7%)	1.0961±1e-3 (0.9%)
GC-MC	0.8090±1e-3	0.7986±5e-4	1.1088±1e-3
+ED	0.7822±2e-3 (3.3%)	0.7848±2e-3 (1.7%)	1.0922±2e-3 (1.5%)
+ND+ED	0.7803±1e-3 (3.5%)	0.7825±1e-3 (2.0%)	1.0901±6e-4 (1.7%)

Table 6: Comparisons of mutual information estimate between learned interaction representations $\{h_{ij}|(i, j) \in \mathcal{E}\}$ and review representations $\{e_{ij}|(i, j) \in \mathcal{E}\}$. The larger score means the smaller divergence.

Models	Digital_Music	Toys_and_Games	Clothing
(1) TransNets	0.09	0.06	0.03
(2) SDNet	0.17	0.14	0.12
(3) RG	0.41	0.37	0.32
(4) RG+ND	1.35	1.22	0.86
(5) RG+ED	2.03	1.92	1.60
(6) RGCL	2.46	2.17	1.82

Moreover, we can obtain a similar conclusion of the strength of ND loss (i.e. β). The best values are $\beta = 0.2$ for 'Digital Music' dataset and $\beta = 0.6$ for 'Clothing' dataset. The choice of β is distinct for different datasets, which depends on how easy the over-fitting happens caused by the specific data size and sparsity.

5.4.2 Generalization of CL Tasks. In order to better measure the generalization of CL tasks, we select review-based NARRE [6] and graph-based GC-MC [37] as backbones. Then, we impose ED on NARRE to boost the interaction modeling. For GC-MC, we impose ND and ED tasks to verify the impact of both ND and ED tasks.

In Table 5, "+ED" denotes imposing ED task into the model, and "+ND+ED" means incorporating two CL tasks into the model. The observations are summarized as follows: First of all, we can observe that both models benefit from our proposed ED and ND tasks, demonstrating the effectiveness and generalization of these two CL tasks. Second, the relative improvement on GC-MC is bigger than the improvement on NARRE across three datasets, demonstrating that ED helps accessing the necessary information during complex interaction modeling. Moreover, when incorporating both ED and ND into GC-MC, the performance of GC-MC can be further improved (3.5%, 2.0%, and 1.7% improvement on three datasets). This phenomenon verifies that ED and ND are essential for learning accurate user preferences.

5.4.3 Quantitative Analysis about the Learned Interaction Features. Previous experiments have proven the impact and generalization of CL tasks. In this section, we intend to make a quantitative analysis to measure the consistency between user-item interaction features and corresponding reviews. Since ED and ND are utilized to help RGCL to learn better node representations, we focus on

the quantitative evaluation of the learned representations. Specifically, following [16], we also leverage MINE [2] to estimate the KL-divergence between the learned user-item interaction $\boldsymbol{h}_{ij}$ and corresponding review representation $\boldsymbol{e}_{ij}$. We have compared the result of TransNets, SDNet, RG+ED, and RG+ND on three datasets, which have been reported in Table 6. Note that the bigger value indicates higher dependence.

According to the results, we can observe the following phenomena: 1) The interaction features generated from RGCL have the best consistency performance, which is consistent with previous experimental results; 2) The comparison between Table 6 (4) and (5) demonstrates that ED has a bigger impact on the model performance than ND, which is consistent with the results in Section 5.4.2; 3) Without CL tasks, our proposed RGCL still performs better than TransNets and SDNet, indicating that our proposed RG is capable of extracting relevant information from reviews and graphs to model user preferences more accurately.

6 CONCLUSION

In order to fully exploit the unique structure of user-item bipartite graph with edge features from ratings and reviews, as well as employ review information to enhance user/item embedding and user-item interaction modeling, we proposed a novel Review-aware Graph Contrastive Learning (RGCL), a graph-based contrastive learning framework for user preference modeling and rating prediction. Specifically, we designed a novel Review-aware Graph learning module (RG) to incorporate review information into user and item embedding learning more effectively. In this component, reviews were utilized to fine-tune the influences of corresponding neighbors and reviews themselves. Moreover, we developed two additional CL tasks (i.e., ED and ND) to constrain RGCL for better node embeddings and interaction modeling. Finally, we have conducted extensive experiments over five benchmark datasets in recommendation to demonstrate the superiority and effectiveness of RGCL. In the future, we will incorporate more advanced review embedding methods for better review representation and design better fusing and alignment methods for complex interactions modeling between reviews and users (items).

ACKNOWLEDGMENTS

REFERENCES

[1] Georgios Alexandridis, Thanos Tagaris, Giorgos Siolas, and Andreas Stafylopatis. 2019. From Free-Text User Reviews to Product Recommendation Using Paragraph Vectors and Matrix Factorization. In *WWW*. 335–343.

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. 2018. Mutual Information Neural Estimation. In *ICML*, Vol. 80. 531–540.

[3] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet Allocation. *JMLR* (2003), 993–1022.

[4] Jiangxia Cao, Xixun Lin, Shu Guo, Luchen Liu, Tingwen Liu, and Bin Wang. 2021. Bipartite Graph Embedding via Mutual Information Maximization. In *WSDM*. 635–643.

[5] Rose Catherine and William Cohen. 2017. TransNets: Learning to Transform for Recommendation. In *RecSys*. 288–296.

[6] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-Level Explanations. In *WWW*. 1583–1592.

[7] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2019. Revisiting Graph Based Collaborative Filtering: A Linear Residual Graph Convolutional Network Approach. In *AAAI*. 27–34.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[9] Xu Chen, Yongfeng Zhang, Hongteng Xu, Zheng Qin, and Hongyuan Zha. 2019. Adversarial Distillation for Efficient Recommendation with External Knowledge. *TOIS* 37, 1 (2019), 1–28.

[10] Xin Dong, Jingchao Ni, Wei Cheng, Zhengzhang Chen, Bo Zong, Dongjin Song, Yanchi Liu, Haifeng Chen, and Gerard de Melo. 2020. Asymmetrical Hierarchical Networks with Attentive Interactions for Interpretable Review-based Recommendation. In *AAAI*. 7667–7674.

[11] Jingyue Gao, Yang Lin, Yasha Wang, Xiting Wang, Zhao Yang, Yuanduo He, and Xu Chu. 2020. Set-Sequence-Graph: A Multi-View Approach Towards Exploiting Reviews for Recommendation. In *CIKM*. 395–404.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *NeurIPS*. 2672–2680.

[13] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *ICML*, Vol. 119. 4116–4126.

[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, YongDong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *SIGIR*. 639–648.

[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[16] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.

[17] Dongmin Hyun, Chanyoung Park, Min-Chul Yang, Ilhyeon Song, Jung-Tae Lee, and Hwanjo Yu. 2018. Review Sentiment-Guided Scalable Deep Recommender System. In *SIGIR*. 965–968.

[18] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[19] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

[20] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[21] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.

[22] Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *ICML*. 1188–1196.

[23] Zeyu Li, Wei Cheng, Reema Kshetramade, John Houser, Haifeng Chen, and Wei Wang. 2021. Recommend for a Reason: Unlocking the Power of Unsupervised Aspect-Sentiment Co-Extraction. In *EMNLP*. 763–778.

[24] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. DAML: Dual Attention Mutual Learning between Ratings and Reviews for Item Recommendation. In *SIGKDD*. 344–352.

[25] Julian McAuley and Jure Leskovec. 2013. Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In *RecSys*. 165–172.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *NerulPS*, Vol. 2. 3111–3119.

[27] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP-IJCNLP*. 188–197.

[28] Francisco J. Peña, Diarmuid O'Reilly-Morgan, Elias Z. Tragos, Neil Hurley, Erika Duriakova, Barry Smyth, and Aonghus Lawlor. 2020. Combining Rating and Review Data by Initializing Latent Factor Models with Topic Models for Top-N Recommendation. In *RecSys*. 438–443.

[29] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *NeurIPS*. 1257–1264.

[30] 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 *ESWC*, Vol. 10843. 593–607.

[31] Sungyong Seo, Jing Huang, Hao Yang, and Yan Liu. 2017. Interpretable Convolutional Neural Networks with Dual Local and Global Attention for Review Rating Prediction. In *RecSys*. 297–305.

[32] Chuan Shi, Xiaotian Han, Li Song, Xiao Wang, Senzhang Wang, Junping Du, and Philip S. Yu. 2021. Deep Collaborative Filtering with Multi-Aspect Information in Heterogeneous Networks. *TKDE* 33, 4 (2021), 1413–1425.

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

[34] Peijie Sun, Le Wu, Kun Zhang, Yanjie Fu, Richang Hong, and Meng Wang. 2020. Dual Learning for Explainable Recommendation: Towards Unifying User Preference Prediction and Review Generation. In *WWW*. 837–847.

[35] Peijie Sun, Le Wu, Kun Zhang, Yu Su, and Meng Wang. 2021. An Unsupervised Aspect-Aware Recommendation Model with Explanation Text Generation. *ACM TOIS* 40, 3, Article 63 (nov 2021).

[36] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-Pointer Co-Attention Networks for Recommendation. In *SIGKDD*. 2309–2318.

[37] Rianne van den Berg, Thomas N. Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *KDD Deep Learning Day* (2017).

[38] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018).

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*. 5998–6008.

[40] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *ICLR*.

[41] Chong Wang and David M. Blei. 2011. Collaborative Topic Modeling for Recommending Scientific Articles. In *SIGKDD*. 448–456.

[42] Chuhan Wu, Fangzhao Wu, Junxin Liu, and Yongfeng Huang. 2019. Hierarchical User and Item Representation with Three-Tier Attention for Recommendation. In *NAACL*. 1818–1826.

[43] Chuhan Wu, Fangzhao Wu, Tao Qi, Suyu Ge, Yongfeng Huang, and Xing Xie. 2019. Reviews Meet Graphs: Enhancing User and Item Representations for Recommendation with Hierarchical Attentive Graph Neural Network. In *EMNLP-IJCNLP*. 4884–4893.

[44] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-Supervised Graph Learning for Recommendation. In *SIGIR*. 726–735.

[45] Le Wu, Lei Chen, Pengyang Shao, Richang Hong, Xiting Wang, and Meng Wang. 2021. Learning Fair Representations for Recommendation: A Graph-based Perspective. In *WWW*. 2198–2208.

[46] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2022. A Survey on Accuracy-oriented Neural Recommendation: From Collaborative Filtering to Information-rich Recommendation. *TKDE* (2022), 1–1.

[47] Libing Yu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. 2019. A Context-Aware User-Item Representation Learning for Item Recommendation. *TOIS* 37, 2 (2019), 22:1–22:29.

[48] Wudong Xi, Ling Huang, Changdong Wang, Yinyu Zheng, and Jianhuang Lai. 2021. Deep Rating and Review Neural Network for Item Recommendation. *TNNLS* (2021), 1–11.

[49] Yonghui Yang, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2021. Enhanced Graph Learning for Collaborative Filtering via Mutual Information Maximization. In *SIGIR*. 71–80.

[50] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *NerulPS*, Vol. 33. 5812–5823.

[51] Junliang Yu, Hongzhi Yin, Min Gao, Xin Xia, Xiangliang Zhang, and Nguyen Quoc Viet Hung. 2021. Socially-Aware Self-Supervised Tri-Training for Recommendation. In *SIGKDD*. 2084–2092.

[52] Chaoyang Li Zeyu Liang, Junping Du. 2020. Abstractive social media text summarization using selective reinforced Seq2Seq attention model. *Neurocomputing* 410 (2020), 432–440.

[53] Kun Zhang, Guangyi Lv, Linyuan Wang, Le Wu, Enhong Chen, Fangzhao Wu, and Xing Xie. 2019. Drr-net: Dynamic re-read network for sentence semantic matching. In *AAAI*. 7442–7449.

[54] Lei Zheng, Vahid Noroozi, and Philip S. Yu. 2017. Joint Deep Modeling of Users and Items Using Reviews for Recommendation. In *WSDM*. 425–434.

[55] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. In *ICML Workshop*.