Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering

Changxin Tian
tianchangxin@ruc.edu.cn
School of Information, Renmin University of China

Yuexiang Xie
yuexiang.xyx@alibaba-inc.com
Alibaba Group

Yaliang Li
yaliang.li@alibaba-inc.com
Alibaba Group

Nan Yang
yangnan@ruc.edu.cn
School of Information, Renmin University of China

Wayne Xin Zhao✉
batmanfly@gmail.com
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing Key Laboratory of Big Data Management and Analysis Methods
Beijing Academy of Artificial Intelligence

ABSTRACT

Recently, graph neural networks (GNN) have been successfully applied to recommender systems as an effective collaborative filtering (CF) approach. However, existing GNN-based CF models suffer from noisy user-item interaction data, which seriously affects the effectiveness and robustness in real-world applications. Although there have been several studies on data denoising in recommender systems, they either neglect direct intervention of noisy interaction in the message-propagation of GNN, or fail to preserve the diversity of recommendation when denoising.

To tackle the above issues, this paper presents a novel GNN-based CF model, named Robust Graph Collaborative Filtering (*RGCF*), to denoise unreliable interactions for recommendation. Specifically, RGCF consists of a graph denoising module and a diversity preserving module. The graph denoising module is designed for reducing the impact of noisy interactions on the representation learning of GNN, by adopting both a hard denoising strategy (*i.e.,* discarding interactions that are confidently estimated as noise) and a soft denoising strategy (*i.e.,* assigning reliability weights for each remaining interaction). In the diversity preserving module, we build up a diversity augmented graph and propose an auxiliary self-supervised task based on mutual information maximization (MIM) for enhancing the denoised representation and preserving the diversity of recommendation. These two modules are integrated in a multi-task learning manner that jointly improves the recommendation performance. We conduct extensive experiments on three real-world datasets and three synthesized datasets. Experiment results show that RGCF is more robust against noisy interactions and achieves significant improvement compared with baseline models.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Recommender Systems, Denoising, Graph Neural Networks

ACM Reference Format:
Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao✉. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531889

1 INTRODUCTION

Nowadays, recommender systems have become increasingly important to help users discover items of interest in real-world applications [2, 49]. As a fundamental technique, collaborative filtering (CF) methods [37, 39] have been widely applied, which capture similar user preference or item characteristics for generating suitable recommendations. Various CF methods have been proposed in the literature [1, 13, 35, 46].Recently, graph neural networks (GNN) [10, 15] provide a new technical approach for developing CF methods. By leveraging the user-item interaction graph, GNN-based CF models have achieved substantial improvement [11, 42, 46, 47] on the recommendation performance.

Despite the effectiveness, existing GNN-based CF models are likely to suffer from the noisy user-item interaction records [5, 27, 30, 45]. For example, some imperfect user behaviors, *e.g.,* erroneous/careless preference selection, inevitably introduce noise to observed interaction data [27, 45]. Besides, when the recommendation setting is implicit feedback (explicit ratings or preference are not available), user behavior does not necessarily indicate his/her

✉ Corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531889

real personal preference (*e.g.,* random surfer click or surrogate shopping) [5, 45]. Even worse, malicious users can deliberately insert well-crafted fake interactions to bias the recommender systems to their benefit [9, 56]. Compared with traditional methods (*e.g.,* matrix factorization), GNN-based recommender systems are potentially more vulnerable to such noise in observed interaction data. A major reason is that GNN adopts a message-passing scheme by iteratively aggregating the neighborhood information, which is prone to enlarging the impact of noisy interactions on the representation learning [48]. Therefore, it is vital to enhance the robustness of GNN-based recommender systems against potential noise in data.

Although there have been several studies on denoising recommender systems [27, 45, 51], these methods cannot effectively resolve the noise issue from GNN-based CF methods. The reasons can be given in twofold. Firstly, previous studies [27, 45] seldom consider the direct intervention of noisy interaction in the representation learning process of GNNs (*i.e.,* the message-passing scheme via aggregating neighborhood information). Therefore, even when the noise of a single node itself is reduced to some extent, the global impact of the aggregated noise from neighborhood remains uncontrolled. Secondly, there is a trade-off between *purity* and *diversity* in denoising interaction data. In this work, purity refers to the major preference of users, while diversity refers to a variety of hidden or minor user interests. Existing methods mainly emphasize the effect of data denoising on purity, while they seldom consider the side effect from denoising, which tend to weaken the diversity of recommendation and even lead to information cocoons.

To address the aforementioned challenges, we propose a novel GNN-based collaborative filtering approach by denoising unreliable interactions, named RGCF (Robust Graph Collaborative Filtering). We design two major technical modules in RGCF. First, we directly improve the graph structure learning by identifying more reliable interactions for message passing (*graph denoising module*). Second, we further enhance the diversity of recommendation by correlating the user preferences learned from denoised interaction data and diversity augmented data (*diversity preserving module*). The two modules are integrated via a multi-task learning strategy to jointly improve the robustness and diversity of recommendation. Specifically, for graph denoising, we estimate the reliability degree for every observed interaction according to the structural similarity between a user and an item. We consider both hard (*i.e.,* discarding interactions that are confidently estimated as noise) and soft (*i.e.,* assigning reliability weights to remaining links) denoising strategies. It aims to reduce the effect of suspicious interactions in the graph learning. For diversity preserving, we consider incorporating a diversity augmented view for the interaction graph, and design an auxiliary self-supervised task based on mutual information maximization (MIM) [22, 44, 61]. By modeling the correlation between the two views (either *denoised* or *diversity augmented*), we can enhance the diversity of the denoised user representations by enforcing a balance between purity and diversity.

To validate the effectiveness of the proposed RGCF, we conduct extensive experiments on six real-world or synthesized recommendation datasets from different domains. Experimental results show that RGCF achieves state-of-the-art performance compared to several competitive baseline methods in terms of accuracy and diversity, especially when noisy interactions are explicitly injected into training data. We also demonstrate that the proposed RGCF is effective to adapt to other GNN-based recommender systems, such as NGCF [46], DGCF [47] and LightGCN [11].

Our main contributions are summarized as: (1) To the best of our knowledge, it is the first denoising method tailored for GNN-based recommender systems; (2) We propose RGCF method to alleviate the negative impact of noisy interactions while preserving the diversity of user's interest; (3) Extensive experiments conducted on three real-world datasets and three synthesized datasets demonstrate the effectiveness of our proposed approach.

2 PRELIMINARY

In this section, we introduce the background for this work.

Collaborative Filtering. Collaborative filtering (CF) techniques are widely applied in recommender systems to capture user's preference over items. Following the setting in literature [39, 49], formally, given the user set $\mathcal{U} = \{u\}$ and item set $\mathcal{I} = \{i\}$, the observed user-item interaction data are denoted as a matrix $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, where the element $r_{u,i} = 1$ if there exists an interaction between the user u and item i, otherwise $r_{u,i} = 0$. Based on the user-item interaction data, we can learn hidden embeddings $\boldsymbol{h}$ for users and items by a representation function $g(\cdot)$ as follows:

$$\boldsymbol{h}_u = g(\mathbf{R}, u), \quad \boldsymbol{h}_i = g(\mathbf{R}, i), \tag{1}$$

where $\boldsymbol{h}_u \in \mathbb{R}^d$ and $\boldsymbol{h}_i \in \mathbb{R}^d$ are the learned embeddings of user u and item i respectively, and d is the hidden dimension of embedding. These learned embeddings will be further used to predict the actual preference of a user over candidate items.

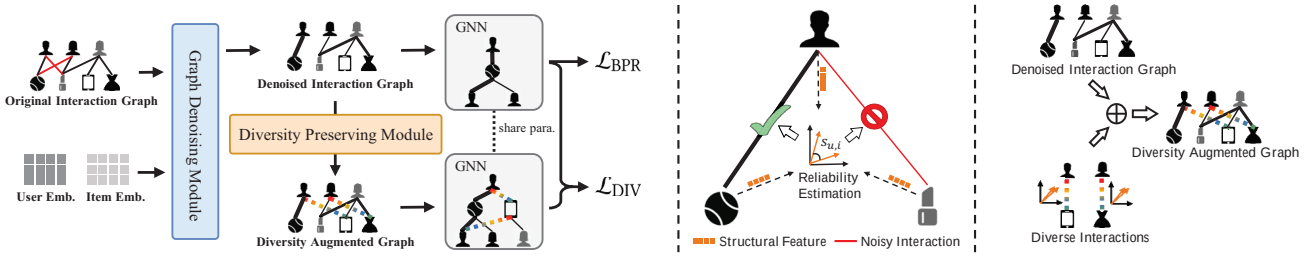
For GNN-based CF, a bipartite interaction graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is constructed based on the user-item interaction data $\mathbf{R}$, where $\mathcal{V}$ consists of the two types of nodes, *i.e.,* user nodes and item nodes, and $\mathcal{E}$ represents their interactions. Specifically, for a nonzero $r_{u,i}$, there is a link between the user node u and item node i in the interaction graph $\mathcal{G}$. To implement the representation function $g(\cdot)$ in Eq. (1), GNN-based CF exploits the graph structure information by applying aggregation functions on the interaction graph $\mathcal{G}$ to produce the embeddings of users and items.

Noise in Recommendation. The concept of *noise* in this work refers to the interactions that cannot reflect real user's preference, including natural noise [45, 51] (*e.g.,* careless preference selection), malicious noise [27, 30] (*e.g.,* the well-crafted fake user-item interactions), and others (*e.g.,* random surfer click or surrogate shopping). Previous works [27, 45] empirically show that the noise degrades the performance of recommender systems. For GNN based CF, the case might become even worse due to the message-passing scheme, which is more vulnerable to noisy user-item interaction data [5, 48].

To tackle such noise issue, we propose a novel GNN-based CF model, which is able to reduce the negative impact of noisy interaction for GNN-based recommendation while preserving the diversity of recommendation to achieve a better performance.

3 METHODOLOGY

In this section, we introduce the proposed Robust Graph Collaborative Filtering (denoted as RGCF), which enhances the robustness of GNN-based collaborative filtering against noisy interactions.



(a) The Overview of RGCF (b) Graph Denoising Module (c) Diversity Preserving Module

Figure 1: (a) RGCF adopts graph denoising module to construct the denoised interaction graph, and further preserves the diversity of recommendation by extracting diversity information from diversity augmented graph. (b) Graph Denoising Module alleviates the impact of noisy interactions by reducing their weights or pruning them based on estimated reliability. (c) Diversity Preserving Module randomly introduces unobserved interactions with high reliability to enrich the denoised graph.

3.1 Overview

The proposed RGCF consists of two major modules, *i.e.,* graph denoising module and diversity preserving module. In specific, the graph denoising module is designed for reducing the effect of noisy user-item interactions in representation learning of GNN-based CF. Based on the estimated reliability degrees of the user-item interactions, we adopt both a hard denoising strategy (*i.e.,* discarding interactions that are confidently estimated as noise) and a soft denoising strategy (*i.e.,* assigning reliability weights to remaining links) in the graph denoising module. As for the diversity preserving module, we enrich the user-item interactions and build up a diversity augmented interaction graph. Then we propose an auxiliary self-supervised task based on mutual information maximization (MIM) to enhance the representation learned from denoised module for preserving the diversity of recommendation. We integrate these two modules via a multi-task learning strategy in RGCF. The overview of RGCF is illustrated in Figure 1, and more details will be introduced in the following sections.

3.2 Graph Denoising Module

In this section, we introduce the graph denoising module in detail.

3.2.1 Reliability Degrees of Interactions. The homophily theory [26, 57] in social networks states that nodes with similar features or structural roles are more likely to interact with each other than those with different features or structural roles. Since node features might be unavailable in CF, we estimate the reliability degrees of the observed interactions between users and items based on their structural similarity in the interaction graph.

Specifically, given the user-item interaction data $\mathbf{R} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, we can construct the interaction graph $\mathcal{G}$, where each nonzero entry $r_{u,i}$ in $\mathbf{R}$ corresponds to an edge between user u and item i in $\mathcal{G}$. Then we extract the one-hop neighbors of nodes (each node might represent a user or an item) as their structural features, which encode the local topology information of nodes. The embedding matrices $\mathbf{E}_U \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $\mathbf{E}_I \in \mathbb{R}^{|\mathcal{I}|\times d}$ are set to transform the user and item structural features into the same hidden space:

$$\mathbf{H}_U^s = \mathbf{R}\mathbf{E}_I, \quad \mathbf{H}_I^s = \mathbf{R}^\top \mathbf{E}_U, \tag{2}$$

where $\mathbf{H}_U^s \in \mathbb{R}^{|\mathcal{U}|\times d}$ and $\mathbf{H}_I^s \in \mathbb{R}^{|\mathcal{I}|\times d}$ are the structural feature matrices of users and items, respectively.

In order to estimate the *reliability degree* of interaction between user u and item i, we measure their structural similarity in the interaction graph $\mathcal{G}$. Given the structural features $\boldsymbol{h}_u^s$ and $\boldsymbol{h}_i^s$, which are the u-th row of $\mathbf{H}_U^s$ and the i-th row of $\mathbf{H}_I^s$, respectively, we employ the cosine similarity function for similarity measurement:

$$\cos\left(\boldsymbol{h}_u^s, \boldsymbol{h}_i^s\right) = \frac{\boldsymbol{h}_u^{s\top}\boldsymbol{h}_i^s}{\|\boldsymbol{h}_u\|_2 \cdot \|\boldsymbol{h}_v\|_2}. \tag{3}$$

Note that the similarity in Eq. (3) cannot be directly used as the weight of edge in $\mathcal{G}$, since it might result in a negative value. Thus we convert the cosine similarity to the normalized cosine distance as the reliability degree $s_{u,i}$, which is formally given as:

$$s_{u,i} = \left(\cos\left(\boldsymbol{h}_u^s, \boldsymbol{h}_i^s\right) + 1\right)/2. \tag{4}$$

The estimated reliability degrees are different from the importance weights defined in previous studies [18, 43], since the reliability degrees are measured based on structural similarity inspired by the homophily theory [26], rather than the attention mechanisms. A large reliability degree score $s_{u,i}$ indicates a reliable interaction between user u and item i according to their structural similarity.

3.2.2 Denoised Interaction Graph. Based on the estimated reliability degrees of user-item interactions, we apply both a soft denoising strategy and a hard denoising strategy on the interaction graph $\mathcal{G}$, which results a denoised interaction graph, denoted as $\widetilde{\mathcal{G}}$.

Specifically, for each observed interaction entry $\langle u, i\rangle$, the graph denoising module utilizes the learned reliability degree $s_{u,i}$ in Eq. (4) as the denoising weights. Furthermore, for the interactions that are regarded as noise with high confidence (*i.e.,* with low reliability degrees), the graph denoising module employs a hard denoising strategy by directly removing them in the message-passing operation of GNN. For each observed interaction $r_{u,i}$, with both soft and hard denoising strategies, the corresponding denoised interaction weight $\widetilde{r}_{u,i}$ can be formally given as:

$$\widetilde{r}_{u,i} = \mathbb{I}(s_{u,i} > \beta) \cdot s_{u,i} \tag{5}$$

where $\mathbb{I}(\cdot)$ is a binary indicator function returning 1 when the condition is true, and β is a hyperparameter that represents the pre-defined threshold value. When the estimated reliability degree $s_{u,i}$ is less than β, the denoised interaction weight $\widetilde{r}_{u,i}$ is set to 0, which implies directly removing the interactions that are confidently estimated as noise.

By re-assigning the interaction weights, we can obtain a denoised interaction matrix $\widetilde{\mathbf{R}}$ based on the original binary interaction matrix $\mathbf{R}$, which can further form a denoised interaction graph $\widetilde{\mathcal{G}}$.Note that $\widetilde{\mathcal{G}}$ is a weighted graph, where the weights of edges are calculated according to Eq. (5) and represent the importance of the interactions in the message-passing operation of GNN. In a nutshell, the graph denoising module alleviates the negative impact of noisy interactions in GNN by reducing their weights or pruning them in the interaction graph.

3.2.3 Graph Neural Network for CF. With the denoised interaction graph $\widetilde{\mathcal{G}}$, we adopt GNNs to learn the embeddings of users and items via message-passing scheme, which can be denoted as:

$$\widetilde{\mathbf{H}}_u, \widetilde{\mathbf{H}}_I = \text{GNN}(\mathbf{H}_U^{(0)}, \mathbf{H}_I^{(0)}, \widetilde{\mathcal{G}}), \tag{6}$$

where $\widetilde{\mathbf{H}}_U$ and $\widetilde{\mathbf{H}}_I$ denote the user and item representations respectively, with $\mathbf{H}_U^{(0)}$ and $\mathbf{H}_I^{(0)}$ as initialized embedding matrices. Various GNN-based models [4, 11, 46] can be used to implement Eq. (6). In RGCF, we adopt efficient LightGCN [11] as an instantiation. Concretely, we perform the message passing on the denoised interaction graph $\widetilde{\mathcal{G}}$ to learn the informative representations by aggregating the neighborhood information as:

$$\begin{aligned} \widetilde{\boldsymbol{h}}_u^{(k)} &= \sum_{i \in \mathcal{N}_u} \frac{\widetilde{r}_{u,i}}{\sum_{j \in \mathcal{N}_u} \widetilde{r}_{u,j}} \widetilde{\boldsymbol{h}}_i^{(k-1)}, \\ \widetilde{\boldsymbol{h}}_i^{(k)} &= \sum_{u \in \mathcal{N}_i} \frac{\widetilde{r}_{u,i}}{\sum_{j \in \mathcal{N}_i} \widetilde{r}_{j,i}} \widetilde{\boldsymbol{h}}_u^{(k-1)}, \end{aligned} \tag{7}$$

where $\widetilde{r}_{u,i}$ represents the denoised interaction weight in Eq. (5), $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the one-hop neighbors set of user u and item i in the denoised interaction graph, respectively. And $\widetilde{\boldsymbol{h}}_u^{(k)}$ and $\widetilde{\boldsymbol{h}}_i^{(k)}$ denote the embeddings of the user u and item i after k layers propagation, which are initialized as $\widetilde{\boldsymbol{h}}_u^{(0)} = \boldsymbol{h}_u^s$ and $\widetilde{\boldsymbol{h}}_i^{(0)} = \boldsymbol{h}_i^s$ when $k = 0$.

After K layers graph propagation according to Eq. (7), GNN outputs the final representations, denoted as $\widetilde{\boldsymbol{h}}_u$ and $\widetilde{\boldsymbol{h}}_i$, by combining the learned embeddings at each layer, which can be given as:

$$\widetilde{\boldsymbol{h}}_u = \sum_{k=0}^{K} \alpha_k \widetilde{\boldsymbol{h}}_u^{(k)}, \quad \widetilde{\boldsymbol{h}}_i = \sum_{k=0}^{K} \alpha_k \widetilde{\boldsymbol{h}}_i^{(k)}, \tag{8}$$

where α_k is a hyper-parameter to denote the importance of embedding learned from k-th layer. During the update process, we only aggregate node embeddings through more reliable edges on the denoised interaction graph, which can prevent the prorogation and enlargement of noisy interaction information. Such a way is able to enhance the robustness of GNN-based recommender systems against noise. We will provide empirical evidence in Section 4.2 to confirm the effectiveness of our proposed denoising strategy.

3.2.4 Prediction. Finally, a prediction layer is built upon the learned representations to predict how likely a user u would interact with an item i (*i.e.,* the user will adopt the recommended item). The prediction layer is implemented with the inner product:

$$\hat{y}_{u,i} = \widetilde{\boldsymbol{h}}_u \cdot \widetilde{\boldsymbol{h}}_i, \tag{9}$$

where $\hat{y}_{u,i}$ is the prediction score for recommendation.

To train the graph denoising module, we sample negative instances from the unobserved user-item interactions to construct pair-wise training data and employ Bayesian Personalized Ranking (BPR) loss [35] for optimization:

$$\mathcal{L}_{\text{BPR}} = \sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}_u} \sum_{j \in \mathcal{I},\, j \notin \mathcal{N}_u} -\log \sigma\left(\hat{y}_{u,i} - \hat{y}_{u,j}\right), \tag{10}$$

where $\sigma(\cdot)$ denotes the sigmoid function, $\mathcal{L}_{\text{BPR}}$ promotes the observed user-item interactions (the interacted items are denoted by $\mathcal{N}_u$) to obtain higher prediction scores than the unobserved ones.

3.3 Diversity Preserving Module

A side effect of graph denoising is that it might weaken the recommendation diversity. To tackle this issue, during the training process, we dynamically construct extra diversity augmented graphs based on the denoised interaction graph, and inject the diversity characteristics into the denoised user representations learned in Eq.(8) via auxiliary Mutual Information Maximization (MIM) loss.

3.3.1 Diversity Augmented Graph. To balance accuracy and diversity of recommendation when denoising, our key idea is to introduce diverse but reliable edges into the denoised interaction graph $\widetilde{\mathcal{G}}$ to construct extra diversity augmented graphs. The construction process includes two steps: random sampling and reliability-aware selection. Specifically, first, we uniformly sample a set of uninteracted user-item pairs with the probability ρ:

$$C = \text{Sample}(\{(u,i) | u \in \mathcal{U}, i \in \mathcal{I}, r_{u,i} = 0\} \mid \rho) \tag{11}$$

where $C = \{(u,i)\}$ is the candidate set from which we select user-item pairs to construct extra diversity augmented graphs. Note that the candidate set C is randomly sampled in each training round, which ensures the diversity of interactions to be add.

Then, we select relatively reliable user-item pairs from the candidate set C based on the estimated reliability degree of each candidate pair. To be more specific, we select the user-item pairs with high reliability degrees, which can be defined formally as:

$$\ddot{r}_{u,i} = \mathbb{I}\left(s_{u,i} \in \text{top-}M\,(S)\right) \cdot s_{u,i} \tag{12}$$

where $s_{u,i}$ is the reliability degree between u and i calculated by Eq. (4), $S = \{s_{u,i} | (u,i) \in C\}$ is the set of reliability degrees of candidate pairs, and M is the number of the selected user-item pairs. Note that unlike the graph denoising module, which removes unreliable interactions from fixed observed data, the diversity preserving module selects interactions from dynamic candidate interactions. In this way, the constructed augmented graph $\ddot{\mathcal{G}}$ is endowed with both diversity and reliability for user-item interactions, where the diversity comes from random sampling and the reliability comes from reliability-aware selection.

3.3.2 Diversity Preserving. Given the constructed diversity augmented graph $\ddot{\mathcal{G}}$, we can learn the corresponding node embeddings with GNN (similar to Eq. (6)). However, since we add unobserved interactions in $\ddot{\mathcal{G}}$, the learned embeddings from $\ddot{\mathcal{G}}$ are likely to deviate from those learned from the denoised interaction graph $\widetilde{\mathcal{G}}$ (Section 3.2.3) according to [40, 54]. In order to balance the accuracy of modeling original graph structure and the diversity of modeling user preference, we design an auxiliary self-supervised task based on Mutual Information Maximization (MIM) [32, 43] to pull the two kinds of node representations close and make a trade-off between the two factors. MIM [32, 43] is based on the concept of

mutual information (MI), which measures dependencies between random variables. Formally, given two variables X and Y, mutual information indicates how much information can be obtained from X by observing Y or vice versa. The mutual information between X and Y, denoted as $I(X, Y)$, is defined as:

$$I(X, Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X), \tag{13}$$

where the entropy $H(\cdot)$ measures the level of expected uncertainty in the random variable. When adapting to graph representation learning, the variables X and Y usually correspond to different views of a graph [48]. By maximizing the mutual information of node representations, we can can capture the correlation between different views of graph and identify the most essential information of the original graph.

Specifically, there exist two views of the user-item interaction graph, *i.e.,* the denoised interaction graph $\widetilde{\mathcal{G}}$ and the diversity augmented interaction graph $\ddot{\mathcal{G}}$. Then, we adopt a weight-sharing GNN introduced in Section 3.2.3 to learn user and item representations based on different interaction graphs:

$$\begin{aligned} \widetilde{\mathbf{H}}_U, \widetilde{\mathbf{H}}_I &= \text{GNN}(\mathbf{H}_U^{(0)}, \mathbf{H}_I^{(0)}, \widetilde{\mathcal{G}}), \\ \ddot{\mathbf{H}}_U, \ddot{\mathbf{H}}_I &= \text{GNN}(\mathbf{H}_U^{(0)}, \mathbf{H}_I^{(0)}, \ddot{\mathcal{G}}), \end{aligned} \tag{14}$$

where $\widetilde{\mathbf{H}}_U$ and $\widetilde{\mathbf{H}}_I$ are the denoised representations, and $\ddot{\mathbf{H}}_U$ and $\ddot{\mathbf{H}}_I$ are the diversity augmented representations.

Maximizing mutual information directly is usually intractable. Inspired by recent studies [6, 22, 41], we adopt the contrastive loss, InfoNCE [31], to optimize a lower bound on MI. Specifically, for each user u, the denoised representation $\widetilde{\boldsymbol{h}}_u$ and the diversity augmented representation $\ddot{\boldsymbol{h}}_u$ are considered as *anchor* and *positive sample*, respectively. Furthermore, the diversity augmented representations of other users are regarded as negative samples. Formally, the diversity preserving loss $\mathcal{L}_{\text{DIV}}$ is defined as:

$$\mathcal{L}_{\text{DIV}} = -\sum_{u,v \in \mathcal{U}} \log \frac{\exp\left(f(\widetilde{\boldsymbol{h}}_u, \ddot{\boldsymbol{h}}_u)/\tau\right)}{\exp\left(f(\widetilde{\boldsymbol{h}}_u, \ddot{\boldsymbol{h}}_u)/\tau\right) + \sum_{v \neq u} \exp\left(f(\widetilde{\boldsymbol{h}}_u, \ddot{\boldsymbol{h}}_v)/\tau\right)}, \tag{15}$$

where τ is temperature hyper-parameter and $f(\cdot, \cdot)$ is cosine similarity function. By minimizing the auxiliary loss $\mathcal{L}_{\text{DIV}}$, we can enforce the correlations between user preferences learned from the denoised interaction data and the diversity augmented data.

Note that our core purpose is to enhance the diversity of the denoised representations, while the diversity augmented representations play an auxiliary role in improving the denoised representations. By correlating the two kinds of representations via the MIM loss, the denoised representations will be injected with the diversity characteristics learned from the diversity augmented graph. Instead of simply fusing the two kinds of representations, our solution takes a safer approach to incorporate diversity information into denoised node representations.

3.4 Model Training

To improve the performance of the proposed RGCF , the graph denoising module and diversity preserving module are integrated via a multi-task learning strategy. The BPR loss $\mathcal{L}_{\text{BPR}}$ and the diversity preserving loss $\mathcal{L}_{\text{DIV}}$ are jointly optimized as:

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \lambda_1 \mathcal{L}_{\text{DIV}} + \lambda_2 \|\Theta\|_2^2, \tag{16}$$

where λ_1 and λ_2 are hyperparameters to control the strengths of the diversity preserving loss and L_2 regularization respectively, and $\Theta = \{\mathbf{E}_U, \mathbf{E}_I\}$ is the set of model parameters. RGCF is trained in an end-to-end manner, *i.e.,* the two modules are optimized jointly.

With the learned parameters $\Theta = \{\mathbf{E}_U, \mathbf{E}_I\}$, we can directly perform graph denoising module and graph neural network (defined in Eq. (2)-(8)) to generate the final representations for inference. In practice, the final representation can be generated offline in advance to speed up online inference.

Time and Space Complexity. We adopt LightGCN [11] as the base model (Section 3.2.3), and the major time complexity of RGCF is in threefold: the graph denoising module, the diversity preserving module and loss calculation. For the graph denoising module, the time complexity of each training epoch is $O(dK|\mathcal{E}| + d|\mathcal{E}|)$. For the diversity preserving module, the time complexity is $O(dK(|\mathcal{E}| + M) + d|C|)$, where C is the candidate set (defined in Eq. (11)) which can be be generated offline. For $\mathcal{L}_{\text{BPR}}$ and $\mathcal{L}_{\text{DIV}}$ calculation, the time complexity is $O(d|\mathcal{E}| + dS|\mathcal{N}|)$, where S is the number of negative samples in diversity preserving loss which can be set as $S \ll |\mathcal{N}|$. Thus, the overall time complexity of RGCF is $O(dK(|\mathcal{E}|+M)+d|\mathcal{E}|+d|C|+dS|\mathcal{N}|)$, which remains the same order as the time complexity of LightGCN $O(dK|\mathcal{E}|+d|\mathcal{E}|)$, and is superior to the complexities of recent denoising recommender systems [5, 45]. On the other hand, RGCF does not introduce additional space complexity besides the basic model. In a nutshell, the proposed RGCF is an efficient and effective graph collaborative filtering framework.

3.5 Comparison and Discussion

In this part, we compare the proposed RGCF with existing methods.

Comparison with Existing Denoising Recommender Systems. Denoising recommender systems is an important research topic in the literature, either based on auxiliary data (*e.g.,* micro-video feature)e [23, 34, 60] or solely on interaction data [7, 38, 45, 51]. Our work falls in the second category of research where no auxiliary data is available when denoising. Compared with existing methods [7, 38, 45, 51], our approach has two major technical contributions. First, RGCF directly prevents the propagation of noise in the message-passing scheme of graph representation learning, which is more effective than node-level denoising. Instead of using observed interactions, we estimate the reliability degrees for interactions according to Eq. (4) and then obtain the denoised interactions according to Eq. (5). In Section 4.3.5, we empirically demonstrate that RGCF effectively distinguishes noise from normal interactions. Second, RGCF further enhances the diversity of the denoised node representations by correlating the denoised view with the diversity augmented view of the interaction graph. Diversity is an important factor to consider for conducting effective recommendations [21, 33, 40, 50, 54], which might be weaken by removing the minor preferences of a user. We construct a diversity augmented graph by randomly introducing reliable edges rather than dropping nodes/edges [48], and adopt the MIM technique to inject the diversity information into the denoised node representation.

Table 1: Statistics of the datasets

Datasets	#Users	#Items	#Actions	Sparsity
MovieLens-1M	6,040	3,629	836,478	96.18%
Yelp	45,478	30,709	1,777,765	99.87%
Amazon-Book	58,145	58,052	2,517,437	99.93%

Comparison with Existing Denoising Graph Neural Networks. Recent studies [14, 24, 36, 55] propose to improve the robustness of GNNs by denoising unreliable edges. However, these studies usually require rich node features as additional input, such as the bag-of-words representations of the documents. As a comparison, RGCF is more suited to the general CF scenario, where only user-item interactions are available. Since our approach is developed based on GNN framework, it is easy to be extended to incorporate side node information besides the interaction data.

4 EXPERIMENTS

In this section, we conduct a series of experiments to demonstrate the effectiveness and robustness of the proposed RGCF.

4.1 Experimental Setup

4.1.1 Dataset. We adopt three real-world recommendation datasets for evaluation:

• **MovieLens-1M**[1] has been widely used in previous studies on recommendation, which contains movie ratings.

• **Yelp**[2] is a business recommendation dataset, in which the businesses in the catering industry are reviewed as the items.

• **Amazon-Book**[3] is the largest category in the public Amazon dataset, which contains users' ratings for their purchased books.

To transform numerical ratings into implicit interaction data, we regard the rating scores above three as the positive feedback. For Yelp and Amazon-Book, we only preserve the users and items that have more than fifteen interaction records to ensure the data quality. The statistics of the adopted datasets are summarized in Table 1. For each dataset, we randomly divide the interactions into training set, validation set, and test set with a ratio of 8:1:1. And we randomly sample one negative item for each positive instance (*i.e.,* the user-item pair) in the training set.

Meanwhile, we construct three synthetic datasets via injecting noises into these real-world datasets, which are denoted as **Polluted MovieLens-1M**, **Polluted Yelp**, and **Polluted Amazon-Book**, respectively. Specifically, we randomly replace 5% of the observed user-item interaction records with noise interaction records to construct the synthesized datasets. For each observed interaction $r_{u,i}$ in the original dataset, we sample a real number ϕ from the uniform distribution $U(0, 1)$. If $\phi < 0.05$, we drop this interaction and sample an item v that user u has not interacted with to construct a fake interaction $r_{u,v}$ as the noise injected into the dataset. We only inject noise into the training set and validation set, leaving the test set unchanged to ensure the reliability of the results.

[1]https://grouplens.org/datasets/movielens/1m/
[2]https://www.yelp.com/dataset
[3]https://jmcauley.ucsd.edu/data/amazon/

4.1.2 Baseline Models. We compare the proposed RGCF with the following several baseline methods:

(1) **BPRMF** [35] is a representative Matrix Factorization (MF) -based model which is optimized by the BPR loss.

(2) **AMF** [12] applies adversarial training on BPRMF to enhance its robustness and improve its generalization performance.

(3) **CDAE** [51] is an auto-encode-based model that learns representations of the users and items via denoising auto-encoder.

(4) **ADT** [45] adaptively prunes noisy interactions to improve the quality of training. In our experiments, we adopt ADT equipped with the Truncated Loss over CDAE [51].

(5) **NGCF** [46] adopts GNN layers on the user-item interaction graph to refine user and item representations.

(6) **DGCF** [47] yields several disentangled representations for users and items to improve the recommendation performance.

(7) **LightGCN** [11] simplifies the design of GCN to make it more concise and appropriate for recommendation.

(8) **SGCN** [5] is a framework that improves the robustness of graph neural networks by training the stochastic binary masks. We adopt LightGCN as the backbone under SGCN framework.

(9) **SGL** [48] utilizes self-supervised learning to improve the accuracy and robustness of LightGCN for recommendation. In our experiments, we adopt the SGL equipped with the Edge Dropout.

4.1.3 Evaluation Metrics. We evaluate the top-N recommendation performance in terms of accuracy and diversity. For the accuracy evaluation, we adopt several top-N metrics, including Mean Reciprocal Rank (MRR), Normalized Discounted Cumulative Gain (NDCG), Recall (Recall), and Hit Ratio (HR). And we set $N = 10$ in the experiments. To ensure the reliability of the experimental results, we perform a full ranking with all item candidates, and report the average results across all the users in the test set [58].

For the diversity evaluation, we adopt the Category Coverage (**Coverage**) defined in previous studies [21, 33, 50], which is calculated by the number of item categories covered by top-N recommendations divided by the total number of item categories. A higher Coverage value implies the top-N recommendation list is more diverse. Note that item categories are unavailable during the training process and only used for evaluation.

4.1.4 Implementation Details. For the baseline methods, we either implement them with the provided source code, or reproduce them with `RecBole`[4] [59]. For a fair comparison, we set the embedding size and batch size to 64 and 4096, respectively, and tune the hyperparameters according to the original papers. We adopt the Adam optimizer for both RGCF and baseline methods, and apply an early stopping strategy with the patience of 10 epochs to prevent overfitting. For the proposed RGCF, we set the layer combination coefficient $\alpha_k = \frac{1}{1+K}$, set the regularization coefficient $\lambda_2 = 1e-5$, and tune the number of layers K in {2, 3}. We set $M = \frac{|\mathbf{R}|}{10} = \frac{|C|}{10}$, where $|\mathbf{R}|$ and $|C|$ are the number of observed interactions and the number of sampled uninteracted user-item pairs, respectively. Besides, we tune the learning rate in [1e-5, 1e-3], the pruning threshold β in {0.02, 0.04, ..., 0.1}, the temperature τ in {0.05, 0.1, 0.2}, and the diversity loss coefficient λ_1 in {1e-5, 1e-6, 1e-7, 1e-8}.

[4]https://github.com/RUCAIBox/RecBole

Table 2: The overall performance comparison on three real-world datasets and three synthesized datasets. The best result is bolded and the runner-up is underlined. * indicates the statistical significance for $p < 0.01$ compared to the best baseline.

Dataset	Metric	BPRMF	AMF	CDAE	ADT	NGCF	DGCF	LightGCN	SGCN	SGL	RGCF	Improv.
MovieLens-1M	Recall	0.1767	0.1804	0.1849	0.1862	0.1807	0.1837	0.1872	0.1892	0.1889	**0.1986***	+4.97%
	NDCG	0.2415	0.2460	0.2461	0.2468	0.2461	0.2479	0.2476	0.2501	0.2505	**0.2565***	+2.40%
	HR	0.7164	0.7213	0.7345	0.7376	0.7243	0.7318	0.7376	0.7412	0.7403	**0.7569***	+2.24%
	MRR	0.4237	0.4298	0.4354	0.4364	0.4285	0.4319	0.4275	0.4369	0.4323	**0.4429***	+1.37%
Yelp	Recall	0.0638	0.0654	0.0647	0.0675	0.0669	0.0728	0.0760	0.0766	0.0775	**0.0830***	+7.10%
	NDCG	0.0455	0.0475	0.0463	0.0487	0.0468	0.0519	0.0537	0.0541	0.0555	**0.0617***	+11.17%
	HR	0.1648	0.1702	0.1662	0.1718	0.1722	0.1835	0.1906	0.1916	0.1923	**0.2060***	+7.12%
	MRR	0.0656	0.0671	0.0678	0.0711	0.0671	0.0741	0.0761	0.0765	0.0787	**0.0896***	+13.85%
Amazon-Book	Recall	0.0617	0.0625	0.0680	0.0693	0.0628	0.0747	0.0806	0.0811	0.0868	**0.0934***	+7.26%
	NDCG	0.0437	0.0443	0.0504	0.0513	0.0438	0.0529	0.0578	0.0612	0.0640	**0.0679***	+5.63%
	HR	0.1555	0.1628	0.1688	0.1724	0.1583	0.1795	0.1911	0.1921	0.2011	**0.2155***	+6.76%
	MRR	0.0621	0.0622	0.0741	0.0753	0.0615	0.0735	0.0795	0.0801	0.0859	**0.0937***	+7.92%
Polluted MovieLens-1M	Recall	0.1540	0.1641	0.1643	0.1673	0.1543	0.1541	0.1592	0.1698	0.1616	**0.1948***	+14.72%
	NDCG	0.2081	0.2160	0.2169	0.2196	0.2086	0.2090	0.2111	0.2213	0.2134	**0.2553***	+15.36%
	HR	0.6839	0.7098	0.7074	0.7124	0.6862	0.6861	0.6902	0.7132	0.6968	**0.7511***	+5.31%
	MRR	0.3748	0.3922	0.3910	0.3968	0.3754	0.3784	0.3744	0.3998	0.3810	**0.4439***	+11.03%
Polluted Yelp	Recall	0.0521	0.0588	0.0597	0.0616	0.0546	0.0631	0.0624	0.0681	0.0684	**0.0777***	+13.60%
	NDCG	0.0369	0.0407	0.0430	0.0440	0.0384	0.0447	0.0438	0.0493	0.0488	**0.0574***	+14.11%
	HR	0.1406	0.1499	0.1572	0.1592	0.1469	0.1638	0.1637	0.1722	0.1731	**0.1951***	+12.71%
	MRR	0.0541	0.0614	0.0634	0.0644	0.0563	0.0649	0.0633	0.0701	0.0696	**0.0840***	+19.83%
Polluted Amazon-Book	Recall	0.0533	0.0603	0.0620	0.0650	0.0500	0.0602	0.0669	0.0722	0.0716	**0.0847***	+17.31%
	NDCG	0.0371	0.0421	0.0459	0.0478	0.0353	0.0430	0.0479	0.0525	0.0528	**0.0606***	+14.77%
	HR	0.1380	0.1503	0.1569	0.1644	0.1346	0.1528	0.1665	0.1742	0.1764	**0.1995***	+13.10%
	MRR	0.0529	0.0584	0.0681	0.0705	0.0517	0.0615	0.0676	0.0713	0.0722	**0.0836***	+15.79%

4.2 Overall Performance

The comparison results between RGCF and baseline methods are shown in Table 2. From this table, we can observe that RGCF achieves the best performances on all datasets and outperforms the state-of-the-art methods SGCN and SGL by a large margin. For the MF-based methods, AMF performs better than BPRMF due to the adoption of adversarial training. CDAE and ADT achieve competitive performance on MovieLens-1M datasets but fail on other datasets, which confirms that auto-encoders are more sensitive to data sparsity. Meanwhile, ADT achieves remarkable improvements over CDAE, which verifies the effectiveness of denoising for recommendation. For GNN-based methods, their performances are better than those of other baseline methods, which demonstrates the importance of utilizing the structural information. In particular, SGCN and SGL achieve the best performance among the baseline methods, which indicates that self-supervised learning and interaction selection can improve the performance of GNN-based methods.

In the scenario where the extra noise is injected into the training data, RGCF also yields significant improvement. Compared with the strongest baselines, RGCF achieves around 10% performance improvement on three polluted datasets. For BPRMF and CDAE, the injected noise significantly decreases their performances, while AMF and ADT are relatively less affected than their corresponding base models since they incorporate the denoising mechanism. Meanwhile, it is worth noting that most GNN-based methods are more sensitive to noise compared to MF-based methods and AE-based methods. The reason is that message-passing scheme in GNN enlarges the negative impact of noise. SGCN removes irrelevant neighbors via applying random masks, which can enhance the robustness and achieve better performances than other baselines. SGL aims to mitigate the impact of noise with self-supervised learning, but it fails to tackle the propagation of noise.

Table 3: Performance comparison with the state-of-the-art methods on three datasets in terms of diversity (Coverage).

Dataset	LightGCN	SGCN	SGL	RGCF	Improv.
MovieLens.	0.4726	0.4702	0.4711	**0.4912**	+3.93%
Yelp	0.0272	0.0270	0.0273	**0.0276**	+1.10%
Amazon.	0.0072	0.0069	0.0068	**0.0073**	+1.39%

Moreover, we compare the diversity performance of RGCF and three competitive GNN-based baseline methods, and show the experimental results in Table 3. From this table we can observe that RGCF achieves the best results in terms of diversity on three datasets. According to Table 2 and Table 3, compared with the base model LightGCN, SGL and SGCN achieve better recommendation accuracy but worse recommendation diversity, which verifies the negative impact of focusing on denoising but ignoring the recommendation diversity. In contrast, RGCF preserves the recommendation diversity when performing denoising, since the proposed

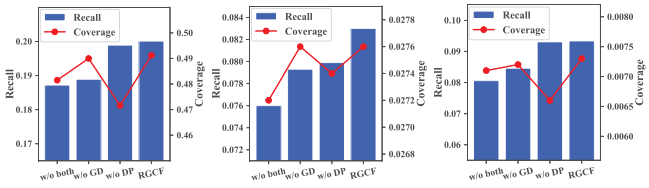
diversity preserving module constructs the augmented graph for enhancing the denoised representations.

By comparing the proposed RGCF with baselines, we conclude that RGCF consistently yields the best performance on six datasets. RGCF not only reduces the negative impact of noisy interactions to enhance the robustness of recommender systems, but also preserves the diversity of recommendation.

4.3 Further Analysis

We conduct a series of detailed analysis on RGCF to further confirm its robustness and effectiveness.

4.3.1 Ablation Study. To validate the contribution of each module in RGCF, we conduct an ablation study on three real-world datasets, i.e., MovieLens-1M, Yelp, and Amazon-Book. We adopt Recall and Category Coverage as the metrics, and present the experimental results in Figure 2, where "*w/o* GD", "*w/o* DP", and "*w/o* both" denote removing graph denoising module, removing diversity preserving module, and removing both of them in RGCF, respectively.



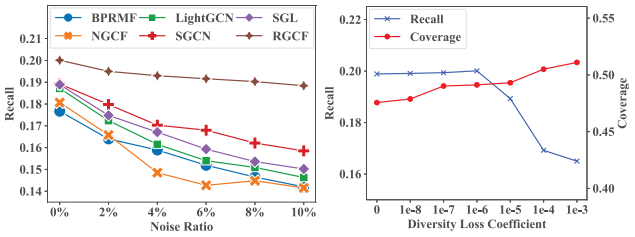
(a) MovieLens-1M (b) Yelp (c) Amazon-Book

Figure 2: Ablation study on three datasets.

From Figure 2, we can observe that the performance of RGCF significantly decreases when graph denoising module or diversity preserving module is removed. Specifically, removing graph denoising module mainly affects the accuracy (Recall), while removing diversity preserving module mainly affects the diversity (Coverage). Meanwhile, the diversity preserving module can also improve the recommendation accuracy, especially for the Yelp dataset. Overall, the ablation study indicates that both of graph denoising module and diversity preserving module contribute to RGCF.

4.3.2 Different Ratios of Noise. We conduct extensive experiments to evaluate the robustness of RGCF with different ratios of noise. Following the dataset construction process described in Section 4.1.1, we pollute the training set and the validation set by replacing a certain ratio of original interactions with random noisy interactions, while keeping the testing set unchanged.

Figure 3(a) shows the experimental results (Recall) on polluted MovieLens-1M dataset. From the figure we can observe that, increasing the ratio of noisy interactions significantly reduces the performances of all baseline methods. The performance degradation of RGCF is much smaller than that of other methods. And the performance of RGCF is consistently better than baseline methods. The gap becomes larger when the ratio of noises increases from 0% to 10%. These observations further confirm the importance of denoising in recommendation, and demonstrate the robustness and effectiveness of RGCF.



(a) Ratio of noise (b) Diversity loss coefficient

Figure 3: Performance comparison w.r.t. different (a) ratios of noise and (b) diversity loss coefficient on MovieLens-1M.

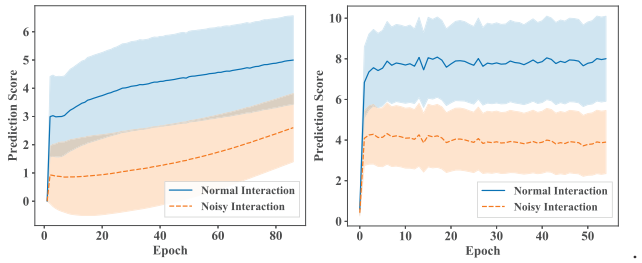
4.3.3 Diversity Loss Coefficient λ_1. In RGCF, the diversity loss coefficient λ_1 is an important hyperparameter to balance the diversity preserving module and the graph denoising module. We vary λ_1 in the range of 0 to 1e-3 to examine the influence of λ_1, and report the Recall and Category Coverage results on MovieLens-1M datasets in Figure 3(b). As shown in Figure 3(b), the diversity performance improves with the increasing of λ_1. However, setting λ_1 to a too large value (*e.g.,* 1e-3) significantly hurts the accuracy performance of RGCF, while an appropriate value can improve both accuracy and diversity. Thus, selecting suitable λ_1 can balance purity and diversity of recommendation when applying RGCF.

4.3.4 Denoising Strategies. In RGCF, we adopt LightGCN as the base GNN model (refer to Section 3.2.3). Actually, the proposed denoising strategies can be applied to various GNN-based CF methods. In this part, we conduct experiments to evaluate whether the proposed denoising strategies can also bring improvements to other GNN-based CF methods. We equip NGCF [46] and DGCF [47] with our proposed denoising strategies, and report the results on three real-world datasets and three polluted datasets in Table 4.

Table 4: Performance comparison of different GNN-based models enhanced by our denoising strategies (Recall). For brevity, R-NGCF and R-DGCF denote the NGCF and DGCF enhanced by our denoising strategies. ML1M represents MovieLens-1M and Ama represents Amazon-Book, while P-ML1M, P-Yelp and P-Ama denote Polluted MovieLens-1M, Polluted Yelp and Polluted Amazon-Book, respectively.

Method	ML1M	Yelp	Ama	P-ML1M	P-Yelp	P-Ama
NGCF	0.1807	0.0669	0.0628	0.1543	0.0546	0.0500
R-NGCF	**0.1886**	**0.0698**	**0.0661**	**0.1766**	**0.0596**	**0.0545**
DGCF	0.1837	0.0728	0.0747	0.1541	0.0631	0.0602
R-DGCF	**0.1892**	**0.0761**	**0.0765**	**0.1759**	**0.0691**	**0.0632**
LightGCN	0.1872	0.0760	0.0806	0.1592	0.0624	0.0669
RGCF	**0.2001**	**0.0830**	**0.0931**	**0.1938**	**0.0777**	**0.0847**

From this table, we can make the following observations: First, with the proposed denoising strategies, the GNN-based CF models achieve better performance compared to their original version, which demonstrates the effectiveness of the proposed denoising strategies. Second, it is worth noting that the enhanced models significantly outperform the original models on polluted datasets



(a) LightGCN **(b) RGCF**

Figure 4: Prediction scores for normal and noisy interactions of RGCF and LightGCN. A larger gap between the prediction scores of noisy and normal interactions implies RGCF performs better than LightGCN in distinguishing noise from normal interactions.

(*e.g.,* P-ML1M, P-Yelp, P-Ama), which is consistent with the finding in Table 2. In addition, LightGCN obtains the greatest improvement compared with other GNN-based CF models, probably due to its simple structure and ease of training. These results further confirm that the proposed denoising strategies can enhance the robustness of GNN-based CF models against the noise.

4.3.5 Prediction Scores of Normal and Noisy Interaction. To further demonstrate that RGCF reduces the negative impact of noisy interactions on the training process, we conduct the analysis by zooming into the training process of RGCF. Specifically, we compare the prediction scores (*i.e.,* $\hat{y}_{u,i}$ in Eq. (9)) of normal interactions and noisy interactions in the training set during the training process of LightGCN and RGCF on Polluted MovieLens-1M, respectively, and report their mean value and standard deviation. The prediction score of a candidate interaction reflects the extent how the model fits the interaction, while a higher score for an interaction indicates that the model fits the interaction better and learns more information from the interaction.

In Figure 4, we can observe that during the training of LightGCN, the prediction scores of noisy interactions gradually increase, and the overlap between the score distribution of noisy and normal interactions becomes larger. This implies that LightGCN has been seriously affected by noisy interactions. In the training process of RGCF, the prediction scores of noisy interactions gradually decrease, and there exists a noticeable gap between the scores of noisy and normal interactions, which indicates that RGCF can effectively distinguish noise from normal interactions.

5 RELATED WORK

In this section, we summarize the related work in two aspects: graph collaborative filtering and denoising recommender systems.

Graph Collaborative Filtering. Unlike traditional collaborative filtering methods [35, 51], graph collaborative filtering models interaction data as a user-item graph and exploits the graph structure for recommendation. Early studies [3, 8, 53] performs random walks in the interaction graph to learn node representations and discover users' preferences. Recently, graph neural networks [10, 15, 44, 52] have been widely adopted in recommender systems [4, 11, 25, 46, 47]. For example, the graph-based auto-encoder framework, GCMC [4], is proposed for explicit matrix completion, NGCF [46] leverages high-order connectivity in user-item graph to improve recommendation, and DGCF [47] learns disentangled representations for users and items. Specially, LightGCN [11] removes nonlinear activation and feature transformation of GCN to simplify the architecture recommendation. Furthermore, UltraGCN [25] simplifies LightGCN by skipping explicit message passing and directly approximating the limit of infinite message passing layers. More recently, self-supervised learning has been introduced into graph collaborative filtering to improve its generalization. SGL [48] devises random data argumentation to construct the auxiliary contrastive task. Despite the remarkable success, they have not considered the diffusion of noise in graph-based methods. Different from previous studies, the proposed RGCF directly reduces weights or even prunes suspicious noisy interactions to alleviate the negative impact in GNN, and improves the accuracy and robustness of GNNs for recommendation.

Denoising Recommender Systems. Significant attention has been dedicated to the robustness of recommender systems, since existing recommendation approaches are vulnerable to various noises [7, 17, 19, 23, 29, 34, 60]. Among existing denoising recommender systems, most studies [23, 34, 60] focus on specific recommendation scenarios, such as micro-video recommendation and next basket recommendation, and perform the denoising with auxiliary data. As for collaborative filtering where only interaction data is available, the early work [29] improves the robustness of recommendation through heuristic neighbor selection. Furthermore, several works [16, 27, 28] adopt probabilistic models to enhance the robustness of factorization-based CF methods against potential noise. Recently, auto-encoder-based models [20, 38, 51] introduce denoising techniques to build more robust recommender systems. Besides, AMF [12] applies adversarial training on the MF model to improve its robustness performance. ADT [45] proposes to denoise implicit feedback for recommendation by discarding the large-loss samples or lowering the weight of large-loss samples. With the development of denoising techniques in graph neural networks [14, 24, 36, 55], the denoising of graph collaborative filtering is also concerned. For instance, SGCN [5] and HAR-GCF [7] reduce the negative effects of noise via stochastic binary mask and attention mechanism, respectively. However, they either apply hard or soft denoising strategy, and there are no consideration on diversity perservation for recommendation.

Summary. To the best of our knowledge, no previous studies are tailored for enhancing the robustness of GNN-based recommender systems, while preserving the diversity of recommendation. While, GNN-based methods are potentially more vulnerable to noise because of their message-passing scheme. Meanwhile, it will impair the diversity of recommendation by only focusing on the denoising. In this paper, we propose RGCF to tackle these two challenges.

6 CONCLUSIONS

In this paper, we propose a novel GNN-based recommendation model, named Robust Graph Collaborative Filtering (RGCF), which integrates a graph denoising module and a diversity preserving

module. We apply two denoising strategies in the graph denoising module to reduce the negative impact of noisy interactions, which are empirically proven effective in GNN-based CF models. Moreover, we construct the diversity augmented graphs to preserve diverse interactions, and design an auxiliary self-supervised task in the diversity preserving module to balance the purity and diversity when denoising. The experimental results show that the proposed RGCF is more robust against noisy interactions and outperforms several competitive baselines by a large margin on six real-world or synthesized recommendation datasets.

In the future, we will further consider extending the current approach strategies to deal with more complex graph structures, such as heterogeneous graphs and dynamic graphs.

ACKNOWLEDGMENTS

This work was partially supported by National Natural Science Foundation of China under Grant No. 61872369, Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. This work was also partially supported by Beijing Academy of Artificial Intelligence (BAAI). Xin Zhao is the corresponding author.

REFERENCES

[1] Fabio Aiolli. 2013. Efficient top-n recommendation for very large scale binary rated datasets. In *Proceedings of the 7th ACM conference on Recommender systems*. 273–280.

[2] Asim Ansari, Skander Essegaier, and Rajeev Kohli. 2000. Internet recommendation systems. *Journal of Marketing Research* 37, 3 (2000), 363–375.

[3] Shumeet Baluja, Rohan Seth, Dharshi Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. 2008. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*. 895–904.

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

[5] Huiyuan Chen, Lan Wang, Yusan Lin, Chin-Chia Michael Yeh, Fei Wang, and Hao Yang. 2021. Structured graph convolutional networks with stochastic masks for recommender systems. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 614–623.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[7] Ping Feng, Yang Qian, Xiaohan Liu, Guoliang Li, and Jian Zhao. 2021. Robust Graph Collaborative Filtering Algorithm Based on Hierarchical Attention. In *International Conference on Web Information Systems and Applications*. 625–632.

[8] Marco Gori, Augusto Pucci, V Roma, and I Siena. 2007. Itemrank: A random-walk based scoring algorithm for recommender engines.. In *IJCAI*, Vol. 7. 2766–2771.

[9] Ihsan Gunes, Cihan Kaleli, Alper Bilge, and Huseyin Polat. 2014. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review* 42, 4 (2014), 767–799.

[10] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[11] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[12] Xiangnan He, Zhankui He, Xiaoyu Du, and Tat-Seng Chua. 2018. Adversarial personalized ranking for recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 355–364.

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 263–272.

[14] Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. 2020. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 66–74.

[15] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations (ICLR '17)*.

[16] Balaji Lakshminarayanan, Guillaume Bouchard, and Cedric Archambeau. 2011. Robust Bayesian matrix factorisation. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 425–433.

[17] Shyong K Lam and John Riedl. 2004. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*. 393–402.

[18] John Boaz Lee, Ryan A Rossi, Sungchul Kim, Nesreen K Ahmed, and Eunyee Koh. 2019. Attention models in graphs: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 6 (2019), 1–25.

[19] Bo Li, Yining Wang, Aarti Singh, and Yevgeniy Vorobeychik. 2016. Data poisoning attacks on factorization-based collaborative filtering. *Advances in neural information processing systems* 29 (2016).

[20] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*. 689–698.

[21] Yile Liang, Tieyun Qian, Qing Li, and Hongzhi Yin. 2021. Enhancing Domain-Level and User-Level Adaptivity in Diversified Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 747–756.

[22] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW*.

[23] Yiyu Liu, Qian Liu, Yu Tian, Changping Wang, Yanan Niu, Yang Song, and Chenliang Li. 2021. Concept-Aware Denoising Graph Neural Network for Micro-Video Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1099–1108.

[24] Dongsheng Luo, Wei Cheng, Wenchao Yu, Bo Zong, Jingchao Ni, Haifeng Chen, and Xiang Zhang. 2021. Learning to drop: Robust graph neural network via topological denoising. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 779–787.

[25] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[26] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[27] Bhaskar Mehta, Thomas Hofmann, and Wolfgang Nejdl. 2007. Robust collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems*. 49–56.

[28] Bhaskar Mehta and Wolfgang Nejdl. 2008. Attack resistant collaborative filtering. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 75–82.

[29] Michael P O'Mahony, Neil J Hurley, and Guénolé CM Silvestre. 2004. Efficient and secure collaborative filtering through intelligent neighbour selection. In *ECAI*, Vol. 16. 383.

[30] Michael P O'Mahony, Neil J Hurley, and Guénolé CM Silvestre. 2006. Detecting noise in recommender system databases. In *Proceedings of the 11th international conference on Intelligent user interfaces*. 109–115.

[31] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[32] Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*. 259–270.

[33] Shameem A Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. 2016. A coverage-based approach to recommendation diversity on similarity graph. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 15–22.

[34] Yuqi Qin, Pengfei Wang, and Chenliang Li. 2021. The world is binary: Contrastive learning for denoising next basket recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 859–868.

[35] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI*. AUAI Press, Arlington, Virginia, USA, 452–461.

[36] Samuel Rey, Santiago Segarra, Reinhard Heckel, and Antonio G Marques. 2021. Untrained Graph Neural Networks for Denoising. *arXiv:2109.11700* (2021).

[37] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, 291–324.

[38] Florian Strub and Jeremie Mary. 2015. Collaborative filtering with stacked denoising autoencoders and sparse inputs. In *NIPS workshop*.

[39] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009).

[40] Jianing Sun, Wei Guo, Dengcheng Zhang, Yingxue Zhang, Florence Regol, Yaochen Hu, Huifeng Guo, Ruiming Tang, Han Yuan, Xiuqiang He, et al. 2020. A framework for recommending accurate and diverse items using bayesian graph convolutional neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2030–2039.

[41] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. 2019. On Mutual Information Maximization for Representation Learning. In *International Conference on Learning Representations*.
[42] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph Convolutional Matrix Completion. *arXiv preprint arXiv:1706.02263* (2017).
[43] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.
[44] Petar Veličković, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2018. Deep Graph Infomax. In *International Conference on Learning Representations*.
[45] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*. 373–381.
[46] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.
[47] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.
[48] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.
[49] Le Wu, Xiangnan He, Xiang Wang, Kun Zhang, and Meng Wang. 2021. A Survey on Neural Recommendation: From Collaborative Filtering to Content and Context Enriched Recommendation. *arXiv preprint arXiv:2104.13030* (2021).
[50] Qiong Wu, Yong Liu, Chunyan Miao, Binqiang Zhao, Yin Zhao, and Lu Guan. 2019. PD-GAN: adversarial learning for personalized diversity-promoting recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 3870–3876.
[51] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. 2016. Collaborative denoising auto-encoders for top-n recommender systems. In *Proceedings of the ninth ACM international conference on web search and data mining*. 153–162.
[52] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.
[53] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 140–144.
[54] Rui Ye, Yuqing Hou, Te Lei, Yunxing Zhang, Qing Zhang, Jiale Guo, Huaiwen Wu, and Hengliang Luo. 2021. Dynamic Graph Construction for Improving Diversity of Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 651–655.
[55] Minji Yoon, Théophile Gervet, Baoxu Shi, Sufeng Niu, Qi He, and Jaewon Yang. 2021. Performance-Adaptive Sampling Strategy Towards Fast and Accurate Graph Neural Networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2046–2056.
[56] Hengtong Zhang, Changxin Tian, Yaliang Li, Lu Su, Nan Yang, Wayne Xin Zhao, and Jing Gao. 2021. Data Poisoning Attack against Recommender System Using Incomplete and Perturbed Data. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2154–2164.
[57] Xiang Zhang and Marinka Zitnik. 2020. Gnnguard: Defending graph neural networks against adversarial attacks. *Advances in Neural Information Processing Systems* 33 (2020), 9263–9275.
[58] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2329–2332.
[59] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 4653–4664.
[60] Jiawei Zheng, Qianli Ma, Hao Gu, and Zhenjing Zheng. 2021. Multi-view Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-start Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2338–2348.
[61] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.