Towards Robust Recommendation via Decision Boundary-aware Graph Contrastive Learning

Jiakai Tang
Sunhao Dai
Zexu Sun
Gaoling School of Artificial Intelligence,
Renmin University of China
Beijing, China
tangjiakai5704@ruc.edu.cn
{sunhaodai,sunzexu21}@ruc.edu.cn

Xu Chen*
Jun Xu
Gaoling School of Artificial Intelligence,
Renmin University of China
Beijing, China
{xu.chen,junxu}@ruc.edu.cn

Wenhui Yu
Lantao Hu
Peng Jiang
Han Li
Kuaishou Technology
Beijing, China
{yuwenhui07,hulantao}@kuaishou.com
{jiangpeng,lihan08}@kuaishou.com

Abstract

In recent years, graph contrastive learning (GCL) has received increasing attention in recommender systems due to its effectiveness in reducing bias caused by data sparsity. However, most existing GCL models rely on heuristic approaches and usually assume entity independence when constructing contrastive views. We argue that these methods struggle to strike a balance between semantic invariance and view hardness across the dynamic training process, both of which are critical factors in graph contrastive learning.

To address the above issues, we propose a novel GCL-based recommendation framework RGCL, which effectively maintains the semantic invariance of contrastive pairs and dynamically adapts as the model capability evolves through the training process. Specifically, RGCL first introduces decision boundary-aware adversarial perturbations to constrain the exploration space of contrastive augmented views, avoiding the decrease of task-specific information. Furthermore, to incorporate global user-user and item-item collaboration relationships for guiding on the generation of hard contrastive views, we propose an adversarial-contrastive learning objective to construct a relation-aware view-generator. Besides, considering that unsupervised GCL could potentially narrower margins between data points and the decision boundary, resulting in decreased model robustness, we introduce the adversarial examples based on maximum perturbations to achieve margin maximization. We also provide theoretical analyses on the effectiveness of our designs. Through extensive experiments on five public datasets, we demonstrate the superiority of RGCL compared against twelve baseline models. To benefit the research community, we have released our project at https://tangjiakai.github.io/RGCL/.

CCS Concepts

• **Information systems → Recommender systems**.

* Corresponding author.

Keywords

Recommender Robustness; Graph Contrastive Learning; Adversarial Learning

1 Introduction

Recently, the intersection of graph neural networks (GNNs) and recommender systems has emerged as a focal point of research attention in both academia and industry [19]. While GNNs have demonstrated remarkable efficacy in capturing high-order connectivity relationships between users and items through their potent message propagation mechanism [18, 37], the inherent data sparsity within recommendation scenarios introduces unexpected bias in users (e.g., non-active vs. active users) and items (e.g., long-tail vs. popular items) representations, thereby impairing the overall model performance [3, 22].

To mitigate the issue of data sparsity and drawing inspiration from self-supervised learning (SSL), recent works have introduced Graph Contrastive Learning (GCL) into GNN-based algorithms [23, 34, 44]. GCL represents a new learning paradigm that integrates contrastive learning [16] with GNN-based recommenders, simultaneously enhancing the alignment of positive embedding pairs and minimizing the similarity to augmented negative instances. In this way, GCL can effectively alleviate the problem of representation degradation among low-degree nodes. In general, GCL-based recommenders can be classified into two categories based on how to build the contrastive samples: (1) **Hardness-driven methods**. These methods basically aim to construct hard enough samples to challenge original recommender models and provide more difficult knowledge to widen the model vision. The methods in this line mainly differentiate themselves by how to define the hardness and how to build hard enough samples. For example, SGL [36] generates challenging views using various strategies, such as node dropout and edge dropout. (2) **Rationality-driven methods**. These methods aim to maintain the rationality of the constructed samples, that
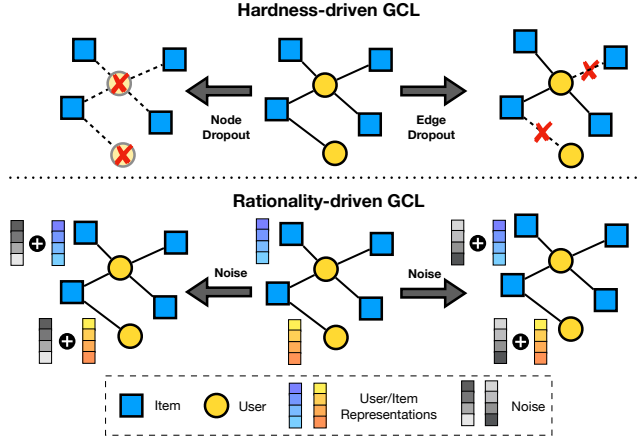
Figure 1: An overview of two types of representative GCL-based recommenders. To facilitate the presentation, we only show a single user and item with injected noise. However, in practice, the semantic-aware GCL-based methods should integrate perturbations to all graph nodes.

is, the augmented features and original labels should form reasonable samples. For example, SimGCL [43] makes slight changes to the original features, such that the augmented feature-label pairs can be still reasonable (*i.e.*, semantically invariant).

Although the aforementioned GCL-based recommenders have shown impressive performance to some extent, we argue that these methods still suffer from several significant limitations. As depicted in Figure 1, **on the one hand**, hardness-driven models blindly pursue the example hardness in contrastive augmentations through manual-designed heuristic strategies. Unfortunately, these models may inadvertently remove certain crucial nodes or edges, neglecting how to maintain task-specific semantics. This oversight makes it challenging for recommenders to accurately capture user preferences and item characteristics. **On the other hand**, rationality-driven methods introduce slight feature perturbations to retain the underlying semantic structure but may overlook the benefits of introducing hard samples on providing more diverse knowledge.

Notably, both challenging positive pairs and hard negative pairs are essential to the success of GCL-based recommenders [30, 32]. In extreme cases, the zero-noise version of contrastive learning may not yield significant performance gains, as verified by prior research [38, 43]. In summary, achieving an adaptive and ideal balance between the hardness and rationality of contrastive augmentations for GCL-based recommenders poses a highly intricate challenge.

In this work, we aim to leverage the idea of *adversarial robustness* [25] to facilitate the construction of optimal contrastive augmented data. To be specific, the goal of adversarial robustness is to promote feature invariance upon task-relevant information, assuring the neural networks are not fooled by imperceptible data perturbations. More importantly, it specifies the maximum perturbation boundary that the current model can tolerate, which explicitly defines a feasible exploration space for conducting example augmentation. Therefore, grounded by such idea, the graph contrastive learning can effectively balance the example hardness

and rationality, both of which are crucial factors to high-quality representations. While this idea is inherently intuitive and holds intriguing potential, its implementation still faces several challenges and obstacles. **C1**: prevalent contrastive augmentation approaches, assuming entity independence, struggle to maintain inherent structural features as they overlook the important connections among user-user and item-item. **C2**: as an unsupervised learning algorithm, GCL in blindly pursuing representation uniformity might unintentionally compromise the robust requirement, that is, narrow margins between data points and the model decision boundary, risking unexpected decreases in the model robustness.

To realize our idea and overcome the above challenges, this paper proposes a novel **R**obust **G**raph **C**ontrastive **L**earning-based recommendation framework, named RGCL. Specifically, we first calculate the maximum perturbation magnitudes for different users and items at each graph layer, while preserving core semantic information for both user and item sides. (**Rationality**) Compared to manual-designed heuristics graph contrastive learning methods, we propose an adversarial-contrastive objective to adaptively generate challenging positive pairs and hard negative pairs based on the global relationships between user-user and item-item, (**Hardness**) which simultaneously overcomes the limitations of the entity independence assumption. (**C1**) At last, we optimize the joint loss of adversarial and contrastive components to concurrently increase the dissimilarity between different users (items) and maximize the distances between user-item inputs and model decision boundary, further improving the robustness of the recommendation model. (**C2**) In summary, our contributions can be summarized as follows:

- We propose a model-agnostic graph contrastive learning framework, which utilizes dynamic decision boundary-aware adversarial perturbations to constrain the perturbation space of contrastive augmented view, achieving a better balance between contrastive hardness and sample rationality.
- We develop a joint learning algorithm based on multi-view contrastive learning and margin maximum adversarial learning to optimize RGCL, empowering better representation uniformity while improving model robustness.
- We give theoretical analyses to underscore the importance of hard contrastive views in model optimization and elucidate the insights behind the efficacy of RGCL in enhancing robustness.
- Extensive experiments on five real-world datasets demonstrate the superior performance of our proposed RGCL framework.

2 Preliminaries

2.1 GNN-based Recommendation

Formally, let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ and $\mathcal{I} = \{i_1, i_2, \ldots, i_N\}$ denote the set of users and items, respectively, where M and N represent the number of users and items, respectively. Considering recommendation scenario with implicit feedback, a binary matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ are typically used to record user-item interactions (*e.g.*, clicks or purchases), where $r_{u,i} = 1$ indicates that user u has interacted with item i, otherwise $r_{u,i} = 0$. Following most GNN-based recommendation works [12, 13, 15], we formulate the interaction behaviors between users and items as a standard bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ involves all graph nodes, and the adjacent matrix

A is defined as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{0}^{M \times M} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0}^{N \times N} \end{bmatrix}.$$

Following the common practice [3, 13], we encode the user u and item i as d-dimensional latent vectors $\mathbf{e}_u \in \mathbb{R}^d$ and $\mathbf{e}_i \in \mathbb{R}^d$, respectively. Besides, $\mathbf{E} = \{\mathbf{e}_u \mid u \in \mathcal{U}\} \cup \{\mathbf{e}_i \mid i \in \mathcal{I}\}$ is defined as the overall learnable embedding matrix for all nodes.

Similar to other GCL-based works [34, 36, 43], this paper adopts the LightGCN [13] as model backbone. Specifically, the comprehensive graph representations $\mathbf{z}_u$ and $\mathbf{z}_i$ for user u and item i in LightGCN are calculated by

$$\mathbf{z}_u = \sum_{l=0}^{L} \mathbf{h}_u^{(l)}, \quad \mathbf{h}_u^{(l)} = \sum_{j \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_j|}} \mathbf{h}_j^{(l-1)}, \quad l \geq 1,$$

$$\mathbf{z}_i = \sum_{l=0}^{L} \mathbf{h}_i^{(l)}, \quad \mathbf{h}_i^{(l)} = \sum_{v \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i||\mathcal{N}_v|}} \mathbf{h}_v^{(l-1)}, \quad l \geq 1,$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ indicate the neighboring nodes of user u and item i, respectively. $\mathbf{h}_u^{(l)}$ and $\mathbf{h}_i^{(l)}$ means the l-th layer graph representation for user u and item i, respectively. Here, $\mathbf{h}_u^{(0)}$ and $\mathbf{h}_i^{(0)}$ are initialized with the learnable embedding $\mathbf{e}_u$ and $\mathbf{e}_i$, respectively. The predicted score $\hat{r}_{u,i}$ for the (u, i) pair is computed as the inner product of their graph representations, i.e., $\hat{r}_{u,i} = \langle \mathbf{z}_u, \mathbf{z}_i \rangle$. Finally, the BPR [29] loss is adopted as the optimization objective:

$$\mathcal{L}_{BPR} = - \sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}_u^+} \sum_{i^- \in \mathcal{I}_u^-} \ln \sigma(\hat{r}_{u,i^+} - \hat{r}_{u,i^-}), \quad (1)$$

where $\sigma(x) = 1/(1 + e^{-x})$, $\mathcal{I}_u^+$ and $\mathcal{I}_u^-$ represent the positive item and unobserved item set for user u, respectively.

2.2 GCL-based Recommenders

In real-world scenarios, interaction behaviors between users and items are actually highly sparse, which can lead to severe overfitting and bias problems [20, 36]. Graph contrastive learning (GCL), as a novel learning paradigm, helps mitigate the above problems [3, 43]. In specific, GCL firstly generates diverse graph views for each user and item (e.g., node dropout and feature masking). Then the different views of the same user (item) are treated as the positive pairs, while the different views of the different instances are treated as the negative pairs. Finally, contrastive learning loss is used to optimize the model parameters with paired users and items, where InfoNCE [27] is the most commonly adopted loss. Formally, the contrastive learning loss for the user side can be defined as follows:

$$\mathcal{L}_{CL}^U(\mathbf{x}_u, \mathbf{y}_u) = \sum_{u \in \mathcal{U}} - \log \frac{\exp(sim(\mathbf{x}_u, \mathbf{y}_u)/\tau)}{\sum_{v \in \mathcal{U}} \exp(sim(\mathbf{x}_u, \mathbf{y}_v)/\tau)}, \quad (2)$$

where $\mathbf{x}_u$ and $\mathbf{y}_u$ denote the two different augmented views of user u, $sim(\cdot, \cdot)$ and τ represents the cosine similarity function and temperature hyper-parameter, respectively. Similarly, the contrastive learning loss of the item side is formulated as follows:

$$\mathcal{L}_{CL}^I(\mathbf{x}_i, \mathbf{y}_i) = \sum_{i \in \mathcal{I}} - \log \frac{\exp(sim(\mathbf{x}_i, \mathbf{y}_i)/\tau)}{\sum_{j \in \mathcal{I}} \exp(sim(\mathbf{x}_i, \mathbf{y}_j)/\tau)}. \quad (3)$$

where $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the two different views of item i.

2.3 Adversarial Robustness

Adversarial training (AT) stands out as one of the most promising approaches for bolstering adversarial robustness [10, 24, 25]. The goal of AT is to increase model robustness by generating adversarial examples through well-designed perturbations, which purposefully induce the neural network to error. Formally, the optimal perturbation for data sample (x, y) is found by maximizing the loss function $\mathcal{L}(\cdot) : \delta^* = \arg\max \mathcal{L}(x + \delta, y; \theta)$ where δ represents an adversarial perturbation of ℓ_p norm smaller than ϵ. Then, the model is trained on a mixture of both original clean examples and generated adversarial examples to enhance the robustness ability.

Discussion. Adversarial robustness uncovers the root cause of the model's adversarial vulnerability, that is, the non-smooth feature space near data samples [17]. In other words, small input perturbations likely result in large changes in the potential semantics, subsequently affecting the model output, which is the basis challenge that adversarial defense algorithms strive to resolve. Actually, this particularly fits well with graph contrastive learning, which aims to maximize the consistency of the given instance under different augmentation views. More importantly, adversarial robustness provides the maximum boundary of feature perturbations that the model can tolerate (cf. Sec 3.2), which effectively restrains the exploration space for contrastive augmentation and guides the generation of optimal view-generator.

3 Our Approach: RGCL

3.1 Overall Framework

The overall framework of RGCL is presented in Figure 2. In specific, we calculate the maximum feature perturbations to guide the subsequent generation of both contrastive examples and adversarial examples. For contrastive examples, we firstly generate two random-augmented views $\mathbf{Z}'$ and $\mathbf{Z}''$ using random perturbations. Besides, the third view $\mathbf{Z}^{ac}$, which we refer to as adversarial-contrastive view, is generated through maximizing relation-aware contrastive function. On the foundation of these contrastive samples, we employ multi-view contrastive learning to prompt high-quality representations. Furthermore, to safeguard the model robustness against potential compromises arising from the uniformity optimization of graph contrastive learning, we generate adversarial examples using maximum perturbation to strenuously enlarge the distances between data points and the decision boundary. Finally, the model is updated by employing a joint optimization objective with augmented contrastive and adversarial data.

3.2 Decision Boundary-aware Perturbation

To build our contrastive samples, we first derive perturbations that the original samples can maximally tolerate to maintain user preferences. Ideally, the perturbations should satisfy two conditions: (1) the perturbations should be as large as possible, such that the obtained contrastive samples are hard enough (**hardness requirement**). (2) The augmented samples after incorporating the perturbations should be still aligned with the user's original preferences (**rationality requirement**).

Different from traditional adversarial learning problems based on classification settings, recommender system is basically a ranking
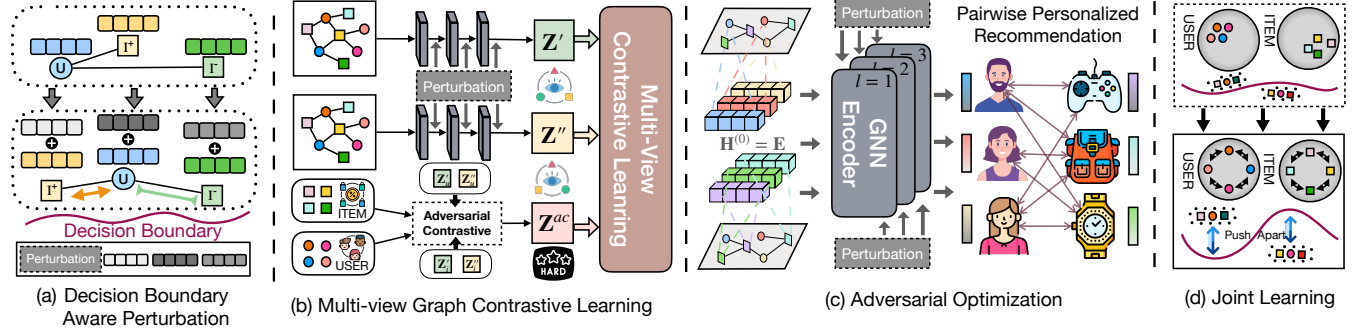
Figure 2: Overall framework of our proposed dynamic decision boundary-aware graph contrastive learning framework RGCL.

problem, and the perturbations should be learned to maintain user preference rankings. To this end, we propose to learn the maximum perturbations that can maintain item pair-wise rankings. Furthermore, given that different orders of graph representations possess different levels of expressive capacity, that is, higher-layer representations aggregate richer structure information and reflect more complex connectivity patterns. Consequently, we tailor the maximum perturbation for each high-order graph representation independently. In specific, for each user u and a positive-negative item pair (i^+, i^-), suppose their original representations are $\mathbf{z}_u = \sum_{l=0}^{L} \mathbf{h}_u^{(l)}$, $\mathbf{z}_{i^+}^{\top} = \sum_{l=0}^{L} \mathbf{h}_{i^+}^{(l)}$, and $\mathbf{z}_{i^-}^{\top} = \sum_{l=0}^{L} \mathbf{h}_{i^-}^{(l)}$, respectively. We define the pair-wise ranking function as $g(u, i^+, i^-) = \left\langle \tilde{\mathbf{z}}_u^{(k)}, \mathbf{z}_{i^+} \right\rangle - \left\langle \tilde{\mathbf{z}}_u^{(k)}, \mathbf{z}_{i^-} \right\rangle$, where $\tilde{\mathbf{z}}_u^{(k)} = \sum_{l=0, l \neq k}^{L} \mathbf{h}_u^{(l)} + (\mathbf{h}_u^{(k)} + \Delta)$ is the user embedding after incorporating perturbation $\Delta \in \mathbb{R}^d$ to k-th layer graph representation $\mathbf{h}_u^{(k)}$, and $< \cdot, \cdot >$ means inner product. Then, the learning objective of perturbation Δ is designed as follows:

$$\Delta_u^{(k)} = \arg\max_{\Delta} ||\Delta||_p \quad \text{s.t. } g(u, i^+, i^-) > 0, \qquad (4)$$

where $\| \cdot \|_p$ means the vector's p-norm. Here, pair-wise ranking function $g(\cdot)$ is linearized around the k-th representation $\mathbf{h}_u^{(k)}$, thus the maximum perturbation $\Delta_u^{(k)}$ is exactly corresponding to the orthogonal projection of $\mathbf{h}_u^{(k)}$ onto the model decision hyperplane.

For the sake of simplicity and better interpretation, we denote that $f(\mathbf{h}_u^{(k)}) = \partial g(u, i^+, i^-) / \partial \mathbf{h}_u^{(k)}$. The maximum perturbation $\Delta_u^{(k)}$ is equivalent to solving for the directional vector from $\mathbf{h}_u^{(k)}$ to the decision boundary, which is formally given as follows:

$$\Delta_u^{(k)} = -\frac{g(u, i^+, i^-)}{\|f(\mathbf{h}_u^{(k)})\|_q^q} \cdot \text{sign}(f(\mathbf{h}_u^{(k)})) \odot \|f(\mathbf{h}_u^{(k)})\|^{q-1}, \quad (5)$$

where $\text{sign}(\cdot)$ is the sign function, and $\odot$ denotes element-wise product. The value of q depends on the choice of perturbation norm ℓ_p ($1 \leq p \leq \infty$), and satisfies that $\frac{1}{p} + \frac{1}{q} = 1$ by following Holder's Inequality's constraint [25]. In our work, p is set as ∞ and q is set as 1, as we empirically found that perturbation constraints under the ℓ_∞ norm have better model performance.

Following that, since users often interact with multiple items in real-world recommendation scenarios, we extend the above method

to all interactions of user u for deriving the final optimal perturbation constraint, which can be rewritten as follows:

$$\Delta_u^{(k)} = -\frac{g(u, i^+, i^-)}{\|f(\mathbf{h}_u^{(k)})\|_1} \cdot \text{sign}(f(\mathbf{h}_u^{(k)})),$$

$$\text{where } i^+, i^- = \arg\min_{i^+ \in \mathcal{I}_u^+, i^- \in \mathcal{I}_u^-} \left| \frac{g(u, i^+, i^-)}{\|f(\mathbf{h}_u^{(k)})\|_1} \right|. \qquad (6)$$

Note that we only focus on perturbing the high-order graph representations for users and items, while skipping the beginning features, i.e., $1 \leq k \leq L$. This is because the original features contain the most abundant semantic information, and polluting these features could lead to a severe performance decrease. On the other hand, by perturbing higher-order representations, we subtly and implicitly disrupt the potential semantic and structural characteristics. Intuitively, it can efficaciously simulates the noise encountered in real-world scenarios, thereby further enhancing the model robustness. Similarly, we can obtain the graph perturbations of item nodes from a dual perspective.

3.3 Relation-aware Contrastive Learning with Perturbation Constraints

As highlighted in Sec. 1, existing GCL-based recommenders struggle to achieve a harmonious balance between contrastive hardness and rationality, both of which are pivotal to acquire high-quality user (item) representations. To this end, in this subsection, we meticulously design the relation-aware adversarial-contrastive objective, which utilizes the global relationships among user-user and item-item to create more challenging positive and hard negative pairs under perturbation constraints. Finally, we optimize the representations through multi-view contrastive learning.

*3.3.1 **Perturbation-constrained Contrastive Augmentation**.*
Following previous works [42, 43], we adopt the random perturbations $\{\mathbf{r}_u^{(l)} : l = 1, 2, \cdots, L\}$ for user u to generate the first random contrastive view $\mathbf{z}_u'$ as follows:

$$\mathbf{z}_u' = \frac{1}{L+1} \left(\mathbf{h}_u^{(0)} + \sum_{l=1}^{L} \left(\mathbf{h}_u^{(l)} + \mathbf{r}_u^{(l)} \right) \right),$$

$$\text{where} \quad \mathbf{r}_u^{(l)} = \epsilon \cdot \frac{\mathbf{r} \odot \text{sign}(\mathbf{h}_u^{(l)})}{\|\mathbf{r} \odot \text{sign}(\mathbf{h}_u^{(l)})\|_2}. \qquad (7)$$

Here, $\mathbf{r} \in \mathbb{R}^d$ following a uniform distribution $U(0, 1)$, and ϵ is a hyper-parameter to control the initial perturbation magnitude. Similarly, we could obtain the augmentation views $\mathbf{z}'_i$ for item i.

Following that, we can get the second augmented representations $\mathbf{z}''_u$ and $\mathbf{z}''_i$ in the same way but utilizing the perturbations $\mathbf{r}$ with different random initialization for more diverse contrastive effects.

However, different users and items have unique *intrinsic robustness*, which means that even imperceptible perturbations may result in large semantic changes for fragile instances. In turn, they unintentionally lead to the erroneous feature-label examples, which is heavily overlooked by existing GCL methods. Therefore, we propose to employ the instance-wise perturbation constrains to guide the generation of contrastive samples, aiming to avoid lossing task-relevant semantic information and build rational view-generator. Specifically, for the l-layer augmentation perturbations $\mathbf{r}_u^{(l)}$, we constrain its exploration space by using the following projection operation $\Pi(\cdot)$ to obtain the constrained perturbation $\tilde{\mathbf{r}}_u^{(l)}$:

$$\tilde{\mathbf{r}}_u^{(l)} = \Pi(\mathbf{r}_u^{(l)}) = \min(abs(\Delta_u^{(l)}), \max(-abs(\Delta_u^{(l)}), \mathbf{r}_u^{(l)}), \quad (8)$$

where $\max(\cdot, \cdot)$ and $\min(\cdot, \cdot)$ are both wise-element operations, and $abs(\cdot)$ computes the absolute value of each element for the given vector. Here, we conservatively constrain the magnitude of random perturbation $\tilde{\mathbf{r}}_u^{(l)}$ within a bounded $\delta_u^{(l)}$-ball, where we define $\delta_u^{(l)}$ as $||\Delta_u^{(l)}||_\infty$. The main motivation behind Eq. (8) is that $\Delta_u^{(l)}$ is the maximum perturbation with the most attacking direction, and our conservative strategy ensures that other perturbation direction bounded within the ball could also safely maintain semantic invariance. Consequently, we replace $\mathbf{r}_u^{(l)}$ in Eq. (7) with constrained perturbation $\tilde{\mathbf{r}}_u^{(l)}$ for achieving contrastive rationality.

3.3.2 *Relation-aware Adversarial-Contrastive Augmentation*. To break the assumption of instance independence in traditional GCL-based algorithms and simultaneously further enhance the hardness of contrastive examples, RGCL generates the relation-aware adversarial-contrastive perturbations to fool the model by confusing the identities among different users and items. To be specific, we propose to maximize the following contrastive loss for generating instance-specific perturbations $\boldsymbol{\eta}$:

$$\max_{\boldsymbol{\eta}} \sum_{u \in \mathcal{U}} -\log \frac{\exp(sim(\ddot{\mathbf{z}}_u, \mathbf{z}''_u)/\tau)}{\exp(sim(\ddot{\mathbf{z}}_u, \mathbf{z}''_u)/\tau) + \sum_{v \in \mathcal{U}/u} \exp(sim(\ddot{\mathbf{z}}_u, \mathbf{z}''_v)/\tau)},$$

$$\text{where } \ddot{\mathbf{z}}_u = \frac{1}{L+1}\left(\mathbf{h}_u^{(0)} + \sum_{l=1}^{L}\left(\mathbf{h}_u^{(l)} + \tilde{\mathbf{r}}_u^{(l)} + \boldsymbol{\eta}_u^{(l)}\right)\right), \quad (9)$$

and $\boldsymbol{\eta} = \{||\boldsymbol{\eta}_u^{(l)}||_\infty \leq \delta_u^{(l)} : u \in \mathcal{U}, l \in \{1, 2, \ldots, L\}\}$ denotes the perturbation set of user u. However, as the general GNN-based recommenders involve nonlinear transformations, it is extremely challenging to find a closed-form solution for the above optimization problem. Drawing inspiration from the fast gradient sign method (FGSM) proposed in Goodfellow *et al.* [10], which assumes that the objective function is approximately linear around the current model parameters. Building on this approximation, we can obtain an optimal max-norm constrained perturbation as follows:

$$\boldsymbol{\eta}_u^{(l)} = \delta_u^{(l)} \cdot \text{sign}(\partial \mathcal{L}_{CL}^U(\ddot{\mathbf{z}}_u, \mathbf{z}''_u)/\partial \boldsymbol{\eta}_u^{(l)}). \quad (10)$$

Similarly, we can compute the relation-aware perturbations for items. Due to space limitation, the detailed derivation steps are omitted here. After that, we generate the relation-aware adversarial-contrastive views for users and items as follows:

$$\mathbf{z}_u^{ac} = \frac{1}{L+1}\left(\mathbf{h}_u^{(0)} + \sum_{l=1}^{L}\left(\mathbf{h}_u^{(l)} + \tilde{\mathbf{r}}_u^{(l)} \odot \text{sign}(\boldsymbol{\eta}_u^{(l)})\right)\right),$$

$$\mathbf{z}_i^{ac} = \frac{1}{L+1}\left(\mathbf{h}_i^{(0)} + \sum_{l=1}^{L}\left(\mathbf{h}_i^{(l)} + \tilde{\mathbf{r}}_i^{(l)} \odot \text{sign}(\boldsymbol{\eta}_i^{(l)})\right)\right),$$

$$(11)$$

where $\tilde{\mathbf{r}}_u^{(l)}$ and $\tilde{\mathbf{r}}_i^{(l)}$ are defined in Eq. (8) and note that they are initialized with different random values.

Compared to the random-augmented view, adversarial-contrastive augmentation has two main advantages: (1) The optimization objective integrates global users (items) to confuse their identities, thus the view generation process is essentially guided by the user-user and item-item relationships, resulting in relation-aware and more challenging contrastive representations. (2) Considering different intrinsic vulnerability among instances, our proposed adversarial-contrastive perturbations are instance-specific and dynamically adopted along with the model training process, thereby further improving the model robustness and adaptability.

3.3.3 *Multi-View Contrastive Learning*. In summary, based on the above discussion, we have obtained views triplets $(\mathbf{z}'_u, \mathbf{z}''_u, \mathbf{z}_u^{ac})$ and $(\mathbf{z}'_i, \mathbf{z}''_i, \mathbf{z}_i^{ac})$ for user u and item i, respectively. Then, we employ multi-view contrastive learning objective for different views of the same instances, *i.e.*, $\{\mathbf{z}'_u \leftrightarrow \mathbf{z}''_u, \mathbf{z}_u^{ac} \leftrightarrow \mathbf{z}'_u, \text{ and } \mathbf{z}_u^{ac} \leftrightarrow \mathbf{z}''_u\}$ for user u, while $\mathbf{z}'_i \leftrightarrow \mathbf{z}''_i, \mathbf{z}_i^{ac} \leftrightarrow \mathbf{z}'_i, \text{ and } \mathbf{z}_i^{ac} \leftrightarrow \mathbf{z}''_i$ for item i.

The complete contrastive loss function is formulated as follows:

$$\mathcal{L}_{CL} = \mathcal{L}_{CL}^U(\mathbf{z}'_u, \mathbf{z}''_u) + \mathcal{L}_{CL}^U(\mathbf{z}_u^{ac}, \mathbf{z}'_u) + \mathcal{L}_{CL}^U(\mathbf{z}_u^{ac}, \mathbf{z}''_u)$$
$$\mathcal{L}_{CL}^I(\mathbf{z}'_i, \mathbf{z}''_i) + + \mathcal{L}_{CL}^I(\mathbf{z}_i^{ac}, \mathbf{z}'_i) + \mathcal{L}_{CL}^I(\mathbf{z}_i^{ac}, \mathbf{z}''_i). \quad (12)$$

where $\mathcal{L}_{CL}^U(\cdot)$ and $\mathcal{L}_{CL}^I(\cdot)$ are defined in Eq. (2) and (3), respectively. Through the multi-view contrastive learning approach, the model is able to acquire more difficult knowledge from hard yet rational contrastive pairs, mitigating recommendation biases and preventing the overfitting resulting from sparse supervised data.

3.4 Towards Margin Maximization via Adversarial Optimization

However, excessive pursuit of representation uniformity in GCL may lead to reduced distances between data points and the decision boundary, potentially compromising the model robustness. We attribute such dilemma is caused by the inherent deficiency that the GCL's essence is unsupervised learning paradigm, which pushes all different instances apart while ignoring task-specific semantic relations [32]. To tackle the above issue, we propose to use adversarial examples for achieving margin maximization. Specifically, we generate adversarial examples using the maximum adversarial perturbation defined in Eq. (6), which can be formulated as follows:

$$\mathbf{z}_u^{adv} = \frac{1}{L+1}\left(\mathbf{h}_u^{(0)} + \sum_{l=1}^{L}\left(\mathbf{h}_u^{(l)} + \Delta_u^{(l)}\right)\right),$$

$$\mathbf{z}_i^{adv} = \frac{1}{L+1}\left(\mathbf{h}_i^{(0)} + \sum_{l=1}^{L}\left(\mathbf{h}_i^{(l)} + \Delta_i^{(l)}\right)\right). \quad (13)$$

We then utilize the generated adversarial examples to optimize the BPR objective (*i.e.*, Eq. (1)), which is given as follows:

$$\mathcal{L}_{ADV} = -\sum_{u \in \mathcal{U}} \sum_{i^+ \in \mathcal{I}_u^+} \sum_{i^- \in \mathcal{I}_u^-} \ln \sigma(\hat{r}_{u,i}^{adv} - \hat{r}_{u,j}^{adv}),$$

$$\text{where } \hat{r}_{u,i}^{adv} = \left\langle \mathbf{z}_u^{adv}, \mathbf{z}_i^{adv} \right\rangle, \ \hat{r}_{u,j}^{adv} = \left\langle \mathbf{z}_u^{adv}, \mathbf{z}_j^{adv} \right\rangle. \tag{14}$$

By explicitly creating adversarial examples around the model's decision boundary, the model optimized with both original and adversarial data can more effectively boost the confidence of input data, thereby enhancing the model's overall robustness.

3.5 Model Training

3.5.1 *Joint Optimization Objective*.
In the training stage, we propose to optimize the model parameters with the joint learning objective, which is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{BPR} + \mu \mathcal{L}_{ADV} + \alpha \mathcal{L}_{CL}, \tag{15}$$

where μ and α are the hyper-parameters for different loss terms.

3.5.2 *Complexity Analysis*.
Since RGCL doesn't introduce any other trainable parameters, the space complexity and the inference time complexity of model remains the same as GNN backbone. Besides, the total training time complexity of RGCL is $O((L|\mathcal{E}|+B^2)d)$, where B and $\mathcal{E}$ denote the batch size and edge set, respectively. Thus, our method retains the same order of computation complexity as other state-of-the-art GCL-based methods, such as SimGCL [43] and RocSE [40]. Due to the limited space, please refer to Appendix A for more detailed analysis.

4 Theoretical Analysis

4.1 Hardness-aware Contrastive Learning

The core motivation of this paper is to construct **semantic preserving** and **hardness enhancing** view-generator for contrastive learning. For the former, we capitalize on the decision boundary-aware constraint to help build rationality-aware views. For the latter, we carefully construct more challenging contrastive paired data because their hardness significantly affects the optimization process of model parameters.

To further explain, we give a proof that contrastive loss is essentially hardness-aware learning mechanism. Specifically, taking the side of users as an example, given a set of users $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$, we denote the similarity of user u_i under different augmented views (*e.g.*, random-augmented view or adversarial-contrastive view) as $s_{i,i}$, and the similarity between user u_i and u_j as $s_{i,j}$. The probability of u_i being identified as u_j is formulated as:

$$P_{i,j} = \frac{\exp(s_{i,j}/\tau)}{\exp(s_{i,i}/\tau) + \sum_{k \neq i} \exp(s_{i,k}/\tau)}.$$

Thus, the objective of contrastive learning is rewritten as follows:

$$\varphi(u_i) = -\log \frac{\exp(s_{i,i}/\tau)}{\exp(s_{i,i}/\tau) + \sum_{k \neq i} \exp(s_{i,k}/\tau)}.$$

Then, the expression of updating model parameters θ is

$$\frac{\partial \varphi(u_i)}{\partial \theta} = \frac{\partial \varphi(u_i)}{\partial s_{i,i}} \frac{\partial s_{i,i}}{\partial \theta} + \sum_{j \neq i} \frac{\partial \varphi(u_i)}{\partial s_{i,j}} \frac{\partial s_{i,j}}{\partial \theta},$$

where we give the derivation results for $\frac{\partial \varphi(u_i)}{\partial s_{i,i}}$ and $\frac{\partial \varphi(u_i)}{\partial s_{i,j}}$:

$$\frac{\partial \varphi(x_i)}{\partial s_{i,i}} = \frac{1}{\tau}(P_{i,i} - 1) \propto \exp(s_{i,i}/\tau),$$
$$\frac{\partial \varphi(u_i)}{\partial s_{i,j}} = \frac{1}{\tau} P_{i,j} \propto \exp(s_{i,j}/\tau), \tag{16}$$

where we can observe that the gradients of the contrastive loss *w.r.t.* both positive and negative pairs are proportional to the corresponding exponential form of their similarity scores. This means that smaller positive pair similarity $s_{i,i}$ and larger negative pair similarity $s_{i,j}$ will have a greater impact on the model parameter optimization. Therefore, our proposed RGCL can learn the high-quality representations by constructing the challenging positive pairs and hard negative pairs, which fits to guide model optimization through hardness-aware contrastive learning.

4.2 Theoretical Analysis of Model Robustness

Although contrastive learning can improve the representation uniformity and reduce the recommendation bias, it may potentially push data points closer to model decision boundary and eventually decrease model robustness due to the nature of task-unrelated unsupervised learning. To make it up, our RGCL explicitly maximizes the margin by constructing adversarial examples based on decision boundary-aware perturbation. Then, in this subsection, we give the explanation on the rationality of our method.

For the sake of notation simplicity, we assume that input example is denoted as x. The goal of recommendation algorithm is to make the preference probabilities for user u's positive items are higher than that for negative items, which is denoted as $g(x; \theta) > 0$. Inspired by work [7], the margin between data point and decision boundary is denoted as $d(x; \theta)$, which can be defined as follows:

$$d(x; \theta) = \|\Delta^*\| = \max \|\Delta\| \quad s.t. \ \Delta : g(x + \Delta; \theta) > 0. \tag{17}$$

We denote the BPR loss function as $\psi(\cdot)$, then we have the theorem:

THEOREM 1. *Gradient descent on $\psi(g(x + \Delta^*; \theta))$ w.r.t. θ with a proper step size increases $d(x; \theta)$, where $\Delta^* = \arg\max_{g(x+\Delta;\theta)>0} \|\Delta\|$ is the maximum perturbation given the current θ.*

PROOF. Let $\rho(\Delta) = \|\Delta\|$ and assume $\rho(\Delta)$ and $\psi(g(x; \theta))$ are functions with twice continuous derivatives in a neighborhood of (Δ^*, θ), c is a constant, and the matrix

$$\begin{pmatrix} \frac{\partial^2 \rho(\Delta^*)}{\partial \Delta^2} + c \cdot \frac{\partial^2 \psi(g(x+\Delta^*;\theta))}{\partial \Delta^2} & \frac{\psi(g(x+\Delta^*;\theta))}{\partial \Delta} \\ \left(\frac{\partial \psi(g(x+\Delta^*;\theta))}{\partial \Delta}\right)^T & 0 \end{pmatrix}$$

is full rank, then we have

$$\nabla d(x; \theta) = C(x, \theta) \frac{\partial \psi(g(x + \Delta^*; \theta))}{\partial \theta},$$

where

$$C(x, \theta) = \frac{\left\langle \frac{\partial \rho(\Delta^*)}{\partial \Delta}, \frac{\partial \psi(g(x+\Delta^*;\theta))}{\partial \Delta} \right\rangle}{\left\| \frac{\partial \psi(g(x+\Delta^*;\theta))}{\partial \Delta} \right\|_2^2}$$

is a scalar. □

The above proof demonstrates that under proper perturbations, our method can maximize the margin by minimizing the adversarial
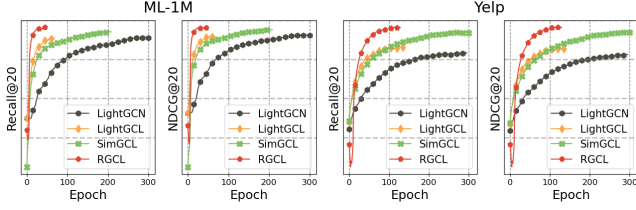
Figure 3: Model convergence analysis w.r.t training epochs on the ML-1M and Yelp datasets.

loss. Therefore, our proposed method can maximize the margin between data points and the model decision boundary by generating adversarial examples with the maximum perturbations defined in Seq. 3.2, thereby effectively improving the robustness of model. Besides, we give an additional robust analysis of our method from the perspective of connections between the sharpness of loss landscape and PAC-Bayes theory. It further theoretically elaborates on the model's tolerance to parameter perturbations. The detailed analysis is presented in the Appendix B.

5 EXPERIMENTS

In this section, we conduct extensive experiments to validate the effectiveness of RGCL, and our goal is to answer the following research questions:

- **RQ1:** How does RGCL perform compared with state-of-the-art recommendation models?
- **RQ2:** How do different designs of RGCL contribute to the final recommendation performance?
- **RQ3:** How does RGCL perform against different data sparsity and item popularity?
- **RQ4:** How do different hyper-parameters affect the recommendation performance of RGCL?

5.1 Experimental Setup

Datasets. We conduct extensive experiments on the following public recommendation datasets: MovieLens (ML)-1M [11], Alibaba [5], Kuaishou [8], Gowalla [6], and Yelp. For detailed introductions and preprocessing details of these datasets, please refer to Appendix C.1.
Baseline Models. We compare RGCL with different state-of-the-art recommendation models, including traditional recommenders (BPR [29] and NeuMF [14]), GNN-based recommenders (GCMC[1], NGCF [33], GCCF [4], and LightGCN [13]) and GCL-based recommenders (GraphCL [41], SGL [36] , LightGCL [3], CGI [34], RocSE [40], and SimGCL [43]). The detailed introduction of all these baseline models are referred to Appendix C.2.
Evaluation Metrics. To ensure the evaluation reliability, following standard practice [34, 36, 39], we adopt the full-ranking strategy to mitigate the evaluation bias introduced by randomly negative sampling, which ranks all the items that are not interacted by the test user as candidate item pool. For evaluation metrics, we adopt the Normalized Discounted Cumulative Gain@K (NDCG@K) and Recall@K, where $K \in \{10, 20, 50\}$.

For better reproducibility, more implementation details are provided in Appendix C.3 and https://tangjiakai.github.io/RGCL/.

5.2 Overall Performance (RQ1)

The results of different methods on all datasets are shown in Table 1. Based on the results, we have the following observations:

- Compared to traditional baselines, such as BPRMF and NeuMF, all GNN-based models perform better on most datasets, which agrees with the previous work and confirms the effectiveness of GNNs [13, 33]. Among all the GNN-based methods, LightGCN usually achieves the excellent performance due to its simple yet effective linear convolution structure. Furthermore, most GCL-based recommenders outperform the GNN-based methods, indicating the desirable property of GCL for alleviating the bias introduced by high-degree nodes. However, these GCL-based models fail to explicitly delineate the definitions of task-relevant semantic rationality and contrastive hardness, thus they achieve inferior balance between contrastive rationality and hardness when constructing augmentation views.

- By comparing our approach with all state-of-the-art baselines, it is clear to see that RGCL yields a consistent boost across all datasets. Besides, the most p-values that are much less than 0.01 also demonstrate the effectiveness of RGCL. We attribute the marked enhancement in performance to the excellent balance between preserving semantic information and bolstering hardness of contrastive examples, which further prompts the ability upper bound of GCL-based recommenders. Besides, we increase the distance between sample points and decision boundary through enhanced adversarial examples, avoiding compromises in robustness caused by contrastive learning.

Training Efficiency. Moreover, to verify the convergence performance of RGCL, we track the Recall@20 and NDCG@20 curves of different models *w.r.t.* the training epochs in Figure 3. From the results, we can observe that RGCL converges significantly faster than SimGCL and LightGCN. Although LightGCL also achieves great convergence speed, its accuracy performance is worse than RGCL, as seen in Table 1. One possible reason is that its static SVD contrastive view fails to keep pace with the evolving model capability during training, eventually limiting the improvement of representation quality. Different from these baselines, RGCL adopts the decision boundary-aware perturbation to guide on the example generation, which adaptively adjusts the augmentation strength to reduce the inconsistency between the representation quality and the contrastive hardness. As a result, RGCL shows both significantly greater efficiency and efficacy.

5.3 Ablation Study (RQ2)

To further validate the importance and contribution of each component in RGCL, we devise multiple simplified variants. In specific, we compare the following four variants: (1) In w/o cons, we drop the decision boundary-aware perturbation constraints on contrastive views. (2) In w/o rand, we do not introduce random initialized perturbation (*i.e.*, set **r** as all-one vector). (3) In w/o ac, we drop the relation-aware view generator but only retain two random augmented views; (4) In w/o adv, we drop the adversarial regularization term $\mathcal{L}_{ADV}$ in the final loss. The experiment is conducted based on the datasets of ML-1M and Yelp, while the observation and conclusion on the other datasets are similar and omitted.
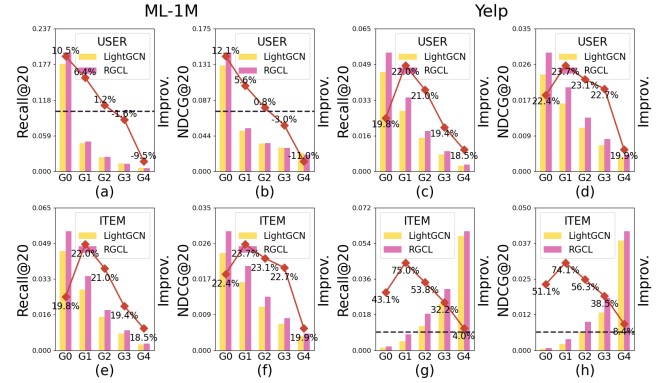
Table 1: Overall performance comparison among baseline and our models. We use bold fonts to label the best performance and use underlines to label the second. The NDCG and Recall metrics are abbreviated as 'N' and 'R', respectively.

Dataset	Metric	BPRMF	NeuMF	GCMC	NGCF	GCCF	LightGCN	GraphCL	SGL	LightGCL	RocSE	CGI	SimGCL	RGCL	Improv.	*p*-value
ML-1M	R@10	0.1702	0.1553	0.1676	0.1763	0.1753	0.1774	0.1837	0.1828	0.1796	0.1786	0.1797	0.1866	**0.1934**	+3.91%	2.67e-4
	N@10	0.2485	0.2291	0.2480	0.2544	0.2624	0.2581	0.2617	0.2625	0.2591	0.2577	0.2613	0.2657	**0.2694**	+1.58%	7.52e-4
	R@20	0.2582	0.2400	0.2526	0.2673	0.2611	0.2680	0.2749	0.2745	0.2722	0.2699	0.2703	0.2798	**0.2901**	+3.69%	7.50e-4
	N@20	0.2576	0.2393	0.2551	0.2647	0.2677	0.2670	0.2721	0.2725	0.2693	0.2676	0.2699	0.2758	**0.2821**	+2.29%	2.26e-3
	R@50	0.4174	0.3952	0.4073	0.4297	0.4171	0.4310	0.4379	0.4381	0.4343	0.4333	0.4308	0.4468	**0.4581**	+2.53%	4.42e-4
	N@50	0.3038	0.2848	0.2985	0.3121	0.3109	0.3137	0.3196	0.3202	0.3162	0.3149	0.3158	0.3242	**0.3321**	+2.42%	4.08e-4
Alibaba	R@10	0.0682	0.0450	0.0503	0.0700	0.0707	0.0734	0.0741	0.0769	0.0747	0.0767	0.0740	0.0791	**0.0824**	+4.20%	1.69e-3
	N@10	0.0435	0.0284	0.0308	0.0446	0.0446	0.0461	0.0473	0.0486	0.0469	0.0485	0.0466	0.0502	**0.0528**	+5.00%	1.57e-4
	R@20	0.1070	0.0718	0.0805	0.1101	0.1104	0.1138	0.1151	0.1187	0.1158	0.1166	0.1146	0.1218	**0.1267**	+4.00%	4.02e-4
	N@20	0.0553	0.0365	0.0399	0.0568	0.0567	0.0584	0.0598	0.0613	0.0594	0.0607	0.0589	0.0632	**0.0663**	+4.85%	1.54e-6
	R@50	0.1875	0.1282	0.1454	0.1920	0.1931	0.1975	0.1944	0.2020	0.2010	0.1937	0.1967	0.2059	**0.2129**	+3.40%	4.63e-4
	N@50	0.0746	0.0501	0.0554	0.0764	0.0765	0.0784	0.0787	0.0812	0.0798	0.0792	0.0786	0.0834	**0.0869**	+4.29%	1.12e-4
Kuaishou	R@10	0.0565	0.0588	0.0645	0.0663	0.0787	0.0730	0.0738	0.0748	0.0775	0.0714	0.0726	0.0788	**0.0899**	+14.14%	5.05e-6
	N@10	0.0326	0.0351	0.0375	0.0370	0.0441	0.0413	0.0436	0.0450	0.0461	0.0409	0.0417	0.0451	**0.0498**	+8.00%	6.99e-4
	R@20	0.0992	0.1095	0.1193	0.1266	0.1327	0.1269	0.1225	0.1282	0.1430	0.1242	0.1316	0.1325	**0.1529**	+6.88%	4.03e-4
	N@20	0.0457	0.0504	0.0541	0.0551	0.0603	0.0573	0.0584	0.0609	0.0660	0.0571	0.0596	0.0613	**0.0687**	+4.09%	3.89e-3
	R@50	0.2027	0.2172	0.2203	0.2562	0.2477	0.2388	0.2366	0.2522	0.2788	0.2489	0.2565	0.2503	**0.2865**	+2.79%	8.94e-3
	N@50	0.0702	0.0760	0.0782	0.0857	0.0879	0.0840	0.0854	0.0902	0.0980	0.0866	0.0891	0.0897	**0.1005**	+2.54%	9.41e-3
Gowalla	R@10	0.1330	0.1205	0.1185	0.1296	0.1319	0.1419	0.1540	0.1470	0.1448	0.1461	0.1447	0.1564	**0.1606**	+2.66%	7.69e-4
	N@10	0.1162	0.1038	0.1013	0.1136	0.1150	0.1257	0.1363	0.1305	0.1277	0.1271	0.1280	0.1379	**0.1419**	+2.89%	1.84e-3
	R@20	0.1894	0.1783	0.1749	0.1878	0.1924	0.2041	0.2178	0.2123	0.2085	0.2117	0.2059	0.2245	**0.2272**	+1.18%	1.83e-2
	N@20	0.1355	0.1238	0.1205	0.1333	0.1356	0.1470	0.1579	0.1527	0.1493	0.1495	0.1487	0.1610	**0.1646**	+2.22%	4.59e-3
	R@50	0.3003	0.2888	0.2832	0.3009	0.3057	0.3194	0.3335	0.3273	0.3240	0.3297	0.3205	0.3460	**0.3468**	+0.23%	1.31e-1
	N@50	0.1682	0.1563	0.1524	0.1667	0.1691	0.1810	0.1922	0.1867	0.1835	0.1845	0.1826	0.1969	**0.2000**	+1.55%	1.58e-3
Yelp	R@10	0.0509	0.0407	0.0520	0.0506	0.0512	0.0612	0.0663	0.0681	0.0626	0.0656	0.0579	0.0740	**0.0753**	+1.75%	1.16e-2
	N@10	0.0392	0.0309	0.0400	0.0390	0.0399	0.0479	0.0518	0.0532	0.0487	0.0512	0.0449	0.0582	**0.0591**	+1.58%	6.58e-3
	R@20	0.0844	0.0691	0.0867	0.0842	0.0851	0.1001	0.1067	0.1098	0.1021	0.1052	0.0940	0.1182	**0.1191**	+0.78%	1.52e-3
	N@20	0.0509	0.0408	0.0520	0.0507	0.0517	0.0614	0.0658	0.0677	0.0624	0.0650	0.0574	0.0736	**0.0744**	+1.09%	2.83e-3
	R@50	0.1571	0.1339	0.1623	0.1570	0.1582	0.1814	0.1909	0.1950	0.1852	0.1871	0.1704	0.2075	**0.2108**	+1.58%	2.36e-3
	N@50	0.0720	0.0596	0.0740	0.0718	0.0730	0.0850	0.0903	0.0925	0.0865	0.0888	0.0796	0.0995	**0.1010**	+1.46%	2.03e-3

We present the results in Table 2, where we can see: For w/o cons variant, unconstrained perturbations result in a significant performance decrease, suggesting that a uniform perturbation cannot effectively preserve that semantic information due to different intrinsic robustness among instances. The w/o rand variant performs much worse than RGCL, which demonstrates that introducing some variances for augmented views is necessary. Furthermore, our method gains improvement over w/o ac variant, which reveals the importance of challenging positive pairs and hard negative pairs However, only optimizing contrastive learning is still sub-optimal, which is evidenced by the lowered performance of w/o adv variant as compared with RGCL. We speculate that over-optimizing contrastive learning for representation uniformity may potentially lead to a reduction in the distance between data points and the model's decision boundary, eventually deteriorating the robustness. In summary, the above observations demonstrate that all the designs are crucial to the final performance improvement.
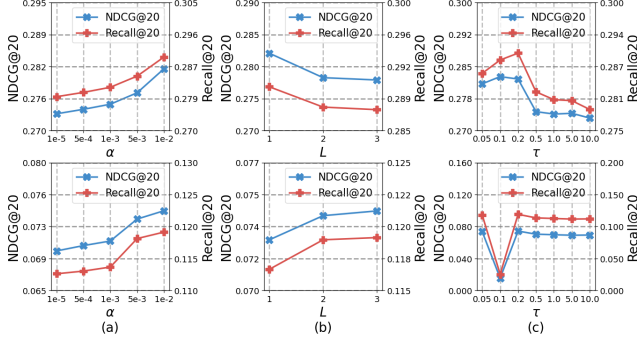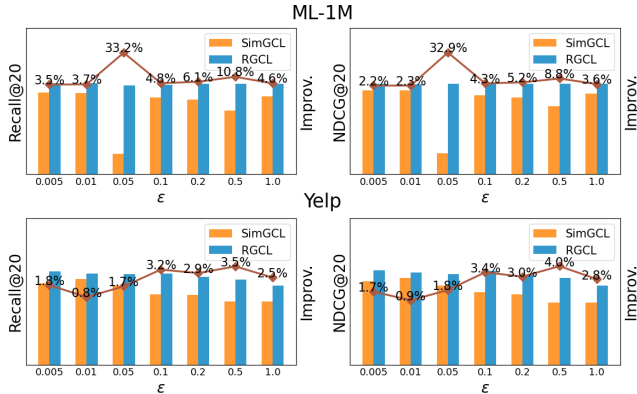
5.4 Robustness Evaluation (RQ3)

To validate the model robustness, we conduct experimental analysis based on different levels of user activity level and item popularity. For detailed user and item grouping approaches, please refer to Appendix C.4. The experimental results are presented in Figure 4, where we can observe that in user (item) groups with sparse interactions, RGCL demonstrates more significant performance

Figure 4: Recommendation performances at different level of data sparsity and item popularity. The black dashed line represents no performance improvement or decline.

improvements. This implies that RGCL effectively capture interest preference of inactive users and characteristic of long-tailed items. Note that the performance trends on the item side for ML-1M and Yelp datasets are different. We speculate that one possible reason is that the proportion of long-tailed items in ML-1M is much higher than Yelp, which results in major contribution to the overall performance by low-degree item groups in ML-1M.

Table 2: Ablation Study on ML-1M and Yelp datasets.

Model	ML-1M				Yelp			
	R@20	N@20	R@50	N@50	R@20	N@20	R@50	N@50
w/o cons	0.2882	0.2798	0.4566	0.3302	0.1185	0.0733	0.2086	0.0995
w/o rand	0.2838	0.2793	0.4470	0.3265	0.1183	0.0736	0.2080	0.0996
w/o ac	0.2872	0.2813	0.4570	0.3315	0.1182	0.0737	0.2085	0.1000
w/o adv	0.2832	0.2801	0.4470	0.3276	0.1180	0.0737	0.2083	0.1000
RGCL	**0.2901**	**0.2821**	**0.4581**	**0.3321**	**0.1191**	**0.0744**	**0.2108**	**0.1010**

Figure 5: Hyper-parameter analysis w.r.t. α, L, τ. The top shows the experimental results on ML-1M and the bottom shows the results on Yelp.

Figure 6: The model tolerance to hyper-parameter ϵ in terms of Recall@20 and NDCG@20 on ML-1M and Yelp datasets. The bars represent the accuracy metrics of different models (w.r.t. NDCG@20 and Recall@20), while the lines show the relative improvement of RGCL compared to SimGCL.

5.5 Further Analysis of RGCL (RQ4)

In this subsection, we further conduct more detailed experiments on the RGCL method to confirm its effectiveness. Due to space limitation, we only show the results on ML-1M and Yelp datasets while the similar conclusions can be derived from other datasets.

5.5.1 Analysis of the model tolerance to hyper-parameter

ϵ. To validate the robustness of our method to perturbation hyper-parameter ϵ, we conduct extensive experiments of performance comparison with SimGCL baseline with different values of ϵ. Specifically, we set the search range as {0.005,0.01,0.05,0.1,0.2,0.5,1.0}. As shown in Figure 6, we observe that SimGCL shows obvious performance fluctuations as ϵ changes. We speculate that the twofold reasons are the following: (1) different instances have different levels of intrinsic robustness. However, uniform and unconstrained perturbations may potentially destroy the semantic structure for fragile instances, ultimately leading to erroneous contrastive views. (2) For instances with better intrinsic robustness, the hardness of contrastive examples is insufficient, hindering the full exploitation of contrastive learning. In contrast, our RGCL adopts decision boundary-aware perturbation constraints to guide the generation of both random and adversarial contrastive examples, leading to stable and superior performance. This demonstrates the insensitivity of RGCL to perturbation hyper-parameter ϵ.

5.5.2 Impact of the coefficient α.

We change α to a set of predetermined representative values presented in Figure 5(a). We can see that the recommendation performance of RGCL gradually improves as α increases, which suggests that contrastive learning can facilitate the uniformity of node representation and learn high-quality features. Correlating with the results in Figure 7 and 8, it also suggests that the personalized characteristic of low-degree users and items can be better captured by our algorithm.

5.5.3 Impact of the layer number L.

To investigate the impact of the GNN layer number on model performance, we vary the hyper-parameter L in the range {1, 2, 3}. From the Figure 5(b), We can observe that the performance trend of RGCL differs across different datasets. For example, for the ML-1M dataset, the over-smoothing issue occurs even with small value of L, while for the Yelp dataset, the model shows the significant performance improvement as graph layer number L increases.

5.5.4 Impact of the temperature τ.

The temperature τ plays an important role in contrastive learning [32]. Figure 5(c) shows the impact of model performance w.r.t. different τ. We can see that the performance fluctuates severely as we use different τ. Specifically, too large values of τ lead to poor performance, which is consistent with the previous work [36]. Conversely, too small temperature values also fail to achieve optimal model performance. One possible reason is that too small τ enforces the model to concentrate few hardest examples that dominate the optimization process, which is

detrimental to achieve the satisfactory generalization ability. Therefore, a suitable temperature is essential to maximize the benefits from graph contrastive learning.

More Analysis. To comprehensively evaluate the superiority of RGCL, we conduct more extensive experiments in Appendix to answer the following research questions:

- **RQ5:** What is the effect of RGCL on improving the representation uniformity of users and items? (*cf.* Appendix D.1)
- **RQ6:** How does the RGCL framework perform when applied to other GNN backbones? (*cf.* Appendix D.2)
- **RQ7:** How does RGCL maintain the semantic information of contrastive examples? (*cf.* Appendix D.3)

6 Related Work

Graph Neural Network in Recommendation. In recent years, the application of GNN models in recommender systems has achieved remarkable success [1, 4, 13, 33]. For example, NGCF [33] models the higher-order connectivity in user-item graph by explicitly injecting collaborative signals into the embedding process. Compared with NGCF, LightGCN [13] simplifies the design of GCN by removing redundant feature transformation and nonlinear activation function. However, GNN-based recommenders suffer from the sparsity of user-item interactions. Although external data sources (*e.g.*, multi-behavior data and knowledge graphs) help mitigate the above issue, obtaining such data is often challenging and even unavailable due to expensive cost or privacy protection. In contrast, graph contrastive learning, as an popular self-supervised learning paradigm, effectively overcomes the challenge of data sparsity.

GCL-based Recommendation Models. Graph contrastive learning (GCL) bridges the advantages of GNN models with contrastive learning, effectively alleviating recommendation bias and simultaneously modeling high-order connectivity. Generally, GCL methods can be classified into hardness-driven models and rationality-driven methods. Specifically, for hardness-driven methods, their key task is to construct diverse and challenging augmented views. For example, GraphCL [41] and SGL [36] both devises multiple heuristic strategy to generate different contrastive views, such edge dropout

and feature masking. However, these methods are prone to losing important semantic features since the augmentation operations are indeed unrelated to the downstream task yet simply based on human-designed experiences. In contrast, rationality-driven GCL methods alleviate the above issue by introducing slight feature perturbations to maintain semantic consistency, such as SimGCL [43] and RocSE [40]. However, these methods still suffer from potential issues, such as insufficient contrastive hardness and tedious trial-and-error of hyper-parameter, resulting in suboptimal performance and poor flexibility. Compared with these methods, our method achieves a better balance between rationality and hardness of contrastive examples via well-designed decision boundary-aware perturbations and adversarial-contrastive view-generator.

7 Conclusion

In this paper, we propose a novel graph contrastive learning framework, named RGCL, aiming to strike a better trade-off between rationality and hardness for the contrastive view-generator. Specifically, we propose a decision boundary-aware perturbation constraints and relation-aware adversarial-contrastive augmentation to generate contrastive examples. Besides, RGCL generates adversarial examples based on the adversarial perturbations to achieve margin maximization between data points and the decision boundary, further improving the model robustness. Finally, we design a joint optimization objective to optimize model parameters.

Acknowledgments

References

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

[2] Zdravko I Botev, Joseph F Grotowski, and Dirk P Kroese. 2010. Kernel density estimation via diffusion. (2010).

[3] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. *arXiv preprint arXiv:2302.08191* (2023).

[4] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 27–34.

[5] Wen Chen, Pipei Huang, Jiaming Xu, Xin Guo, Cheng Guo, Fei Sun, Chao Li, Andreas Pfadler, Huan Zhao, and Binqiang Zhao. 2019. POG: personalized outfit generation for fashion recommendation at Alibaba iFashion. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2662–2670.

[6] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[7] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. 2020. MMA Training: Direct Input Space Margin Maximization through Adversarial Training. In *International Conference on Learning Representations*.

[8] Chongming Gao, Shijun Li, Yuan Zhang, Jiawei Chen, Biao Li, Wenqiang Lei, Peng Jiang, and Xiangnan He. 2022. KuaiRand: An Unbiased Sequential Recommendation Dataset with Randomly Exposed Videos. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3953–3957.

[9] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[11] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[12] Wei He, Guohao Sun, Jinhu Lu, and Xiu Susie Fang. 2023. Candidate-aware Graph Contrastive Learning for Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1670–1679.

[13] 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.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[15] Tinglin Huang, Yuxiao Dong, Ming Ding, Zhen Yang, Wenzheng Feng, Xinyu Wang, and Jie Tang. 2021. Mixgcf: An improved training method for graph neural network-based recommender systems. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 665–674.

[16] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. 2020. A survey on contrastive self-supervised learning. *Technologies* 9, 1 (2020), 2.

[17] Ziyu Jiang, Tianlong Chen, Ting Chen, and Zhangyang Wang. 2020. Robust pre-training by adversarial contrastive learning. *Advances in neural information processing systems* 33 (2020), 16199–16210.

[18] Xuewu Jiao, Weibin Li, Xinxuan Wu, Wei Hu, Miao Li, Jiang Bian, Siming Dai, Xinsheng Luo, Mingqing Hu, Zhengjie Huang, et al. 2023. PGLBox: Multi-GPU Graph Learning Framework for Web-Scale Recommendation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4262–4272.

[19] Di Jin, Luzhi Wang, Yizhen Zheng, Guojie Song, Fei Jiang, Xiang Li, Wei Lin, and Shirui Pan. 2023. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In *Proceedings of the ACM Web Conference 2023*. 684–693.

[20] Mengyuan Jing, Yanmin Zhu, Tianzi Zang, and Ke Wang. 2023. Contrastive self-supervised learning in recommender systems: A survey. *ACM Transactions on Information Systems* 42, 2 (2023), 1–39.

[21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[22] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In *Proceedings of the ACM Web Conference 2022*. 2320–2329.

[23] Lingyun Lu, Bang Wang, Zizhuo Zhang, Shenghao Liu, and Han Xu. 2023. VRKG4Rec: Virtual Relational Knowledge Graph for Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 526–534.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[25] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2574–2582.

[26] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. 2017. Exploring generalization in deep learning. *Advances in neural information processing systems* 30 (2017).

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

[28] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[30] Joshua Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. *arXiv preprint arXiv:2010.04592* (2020).

[31] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[32] Feng Wang and Huaping Liu. 2021. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2495–2504.

[33] 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.

[34] Chunyu Wei, Jian Liang, Di Liu, and Fei Wang. 2022. Contrastive Graph Structure Learning via Information Bottleneck for Recommendation. *Advances in Neural Information Processing Systems* 35 (2022), 20407–20420.

[35] Yuxin Wen, Shuai Li, and Kui Jia. 2020. Towards understanding the regularization of adversarial robustness on neural networks. In *International Conference on Machine Learning*. PMLR, 10225–10235.

[36] 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.

[37] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[38] Jun Xia, Lirong Wu, Jintao Chen, Bozhen Hu, and Stan Z Li. 2022. Simgrace: A simple framework for graph contrastive learning without data augmentation. In *Proceedings of the ACM Web Conference 2022*. 1070–1079.

[39] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-Contrastive Graph Learning for Recommendation. (2023).

[40] Haibo Ye, Xinjie Li, Yuan Yao, and Hanghang Tong. 2023. Towards robust neural graph collaborative filtering via structure denoising and embedding perturbation. *ACM Transactions on Information Systems* 41, 3 (2023), 1–28.

[41] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in neural information processing systems* 33 (2020), 5812–5823.

[42] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2023. XSimGCL: Towards extremely simple graph contrastive learning for recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[43] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Lizhen Cui, and Quoc Viet Hung Nguyen. 2022. Are graph augmentations necessary? simple graph contrastive learning for recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1294–1303.

[44] Chi Zhang, Rui Chen, Xiangyu Zhao, Qilong Han, and Li Li. 2023. Denoising and Prompt-Tuning for Multi-Behavior Recommendation. In *Proceedings of the ACM Web Conference 2023*. 1355–1363.

Table 3: Statistics of the datasets.

Dataset	#Users	#Items	#Interactions	Sparsity
ML-1M	6,038	3,489	820,336	96.1059%
Alibaba	12,265	6,145	193,120	99.7437%
Kuaishou	2,457	1,042	35,795	98.6019%
Gowalla	13,149	14,009	535,650	99.7092%
Yelp	42,324	28,748	1,611,965	99.8675%

Content of Appendix

A Analysis of Training Time Complexity

The extra training time complexity of RGCL comes from the loss terms of contrastive and adversarial components. Suppose the number of nodes and edges are $|\mathcal{V}|$ and $|\mathcal{E}|$, respectively. Let B denote the batch size, d denote the embedding dimension, L denote the total layer number. We analyze the time complexity of each component as follows:

- **Original loss**. The time complexity of the original LightGCN model comes from adjacent matrix construction, graph convolution computation and BPR calculation. Their time complexities are $O(|\mathcal{E}|)$, $O(L|\mathcal{E}|d)$ and $O(Bd)$ respectively. Therefore, the total time complexity is $O((L|\mathcal{E}| + B)d)$.
- **Contrastive loss**. To begin with, solving for the perturbation constraints in contrastive learning needs one pass of forward and backward propagation, where the time complexity is $O(L|\mathcal{E}|d)$. Then, constructing two random-augmented views requires two pass of forward propagation. As for adversarial-contrastive view, it also needs extra one pass of forward and backward propagation, where the time complexity of the contrastive loss paradigm is $O(B^2d)$. Therefore, the total time complexity of the contrastive learning component is $O((L|\mathcal{E}| + B^2)d)$.
- **Adversarial loss**. The adversarial perturbations for generating adversarial examples has already been accounted in the contrastive loss part. Thus, in this part, we simply consider the time complexity of forward propagation and BPR loss, which are $O(L|\mathcal{E}|d)$ and $O(Bd)$, respectively. Therefore, the total time complexity of the adversarial loss is $O((L|\mathcal{E}| + B)d)$.

In summary, the total time complexity of the proposed RGCL is $O((L|\mathcal{E}|+B^2)d)$, which maintains the same order of time complexity as other graph contrastive learning algorithms [40, 43]. However, the experimental results in Figure 3 demonstrates that our algorithm has better converge and accuracy performance.

B Further Robustness ANALYSIS

Inspired by previous work [26, 38], we provide the robustness analysis from the perspective of connections between sharpness of loss landscape and PAC-Bayes theory. Generally, smoother feature space can avoid large feature variations caused by input perturbations [35]. Meanwhile, from the perspective of model optimization, flatter loss landscape can bring better model robustness. Specifically, assuming that the prior distribution Q over the model parameters, with probability at least $1 - \xi$ over the draw of the training data, the expected error of $\mathcal{L}_{BPR}$ can be bounded as follows:

$$\mathbb{E}_\Delta \left[\widetilde{\mathcal{L}}_{BPR} \right] \leq \mathbb{E}_\Delta \left[\mathcal{L}_{BPR} \right] + 4\sqrt{\frac{\text{KL}(\theta + \xi \| Q) + \ln \frac{2m}{\xi}}{m}}, \quad (18)$$

where $\widetilde{\mathcal{L}}_{BPR}$ represents the expected error, m is the size of training data, Δ denotes the perturbation of model parameter. Then, we rewrite the above bound as follows:

$$\mathbb{E}_\Delta \left[\widetilde{\mathcal{L}}_{BPR} \right] \leq \mathbb{E} \left[\mathcal{L}_{BPR} \right] + \underbrace{\mathbb{E}_\Delta \left[\mathcal{L}_{BPR} \right] - \mathbb{E} \left[\mathcal{L}_{BPR} \right]}_{\text{Expected sharpness}}$$
$$+ 4\sqrt{\frac{\text{KL}(\theta + \Delta \| Q) + \ln \frac{2m}{\xi}}{m}}, \quad (19)$$

where expected sharpness $\mathbb{E}_\Delta \left[\mathcal{L}_{BPR} \right] - \mathbb{E} \left[\mathcal{L}_{BPR} \right]$ demonstrates that our method aims to reduce the sensitivity to model parameter variations and increase the smoothness of the feature space. Therefore, the proposed perturbation-based augmentation examples can achieve more robust and well-generalized model performance.

C EXPERIMENT DETAILS

C.1 Recommendation Datasets

We conduct extensive experiments on the following five publicly available recommendation datasets in this paper: (1) **MovieLens (ML)-1M**[1] is a widely adopted movie recommendation dataset, containing the one million movie ratings provided by users, ranging from 1 to 5 stars. (2) **Alibaba**[2] is a fashion-related dataset and provides user behaviors related to both outfits and fashion items. (3) **Kuaishou**[3] contains user interactions on exposed short videos, collected from the video-sharing mobile App. (4) **Gowalla**[4] is a checking-in dataset for item recommendation, collected from a location-based social networking website. (5) **Yelp**[5] is a widely-used business recommendation dataset collected from yelp website, where the business venues of users are viewed as the items.

To transform the explicit user ratings into implicit interaction behavior, the interactions with ratings above three are viewed as the positive example for rating-based datasets (*i.e.*, ML-1M and Yelp). For Yelp and Gowalla datasets, we filter users and items that have less than fifteen interaction number to ensure the data quality. For all datasets, we randomly divide the data into training set, validation set and testing set using a ratio of 8:1:1. For negative samples

---

[1] https://grouplens.org/datasets/movielens/
[2] https://github.com/wenyuer/POG
[3] https://kuairand.com/
[4] https://snap.stanford.edu/data/loc-gowalla.html
[5] https://www.kaggle.com/datasets/yelp-dataset/yelp-dataset/versions/2?resource=download

used in BPR objective, we uniformly sample one negative item for each positive interaction. The overall experiments are repeated five times with different initialized seeds for significance test of model performance. The statistics of the five recommendation datasets are shown in Table 3.

C.2 Baselines

Traditional Recommenders:
- **BPRMF** [29] is a well known matrix factorization model by optimizing BPR loss function.
- **NeuMF** [14] is a deep recommendation model, which aims to capture the non-linear correlations between users and items.

GNN-based Recommenders:
- **GCMC** [1] is a graph auto-encoder framework to learn complex patterns and dependencies within the user-item interaction graph by differentiable message passing.
- **NGCF** [33] is a collaborative filtering model that integrates interactions of user-item bipartite into the embedding process for modeling high-order connectivity.
- **GCCF** [4] is a linear graph recommendation model, which alleviates the over smoothing problem by removing non-linearity and introducing the residual network structure.
- **LightGCN** [13] is a graph-based recommender model, which enhances the collaborative filtering information by abandoning the feature transformation and nonlinear activation.

GCL-based Recommenders:
- **GraphCL** [41] is a graph contrastive learning framework, which designs various types of graph augmentations to incorporate transformation randomness (*e.g.*, attribute masking).
- **SGL** [36] is a self-supervised learning method based on user-item bipartite interaction graph, which devises three augmentation strategies, *aka.*, node dropout, edge dropout and random walk.
- **LightGCL** [3] is a simple graph contrastive paradigm that utilizes the SVD for contrastive augmentation to integrate the global collaborative relation without structural refinement.
- **RocSE** [40] is a robust graph collaborative filtering model, which adds in-distribution perturbation to construct a contrastive view-generator, which mimicking the behaviors of adversarial attacks.
- **CGI** [34] is a graph contrastive model by designing learnable graph augmentation to adaptively learn whether to drop an edge or node and leveraging the information bottleneck technique to guide contrastive learning process.
- **SimGCL** [43] is a GCL-based recommendation model, which discards the sophisticated graph augmentation and adopts to add uniform noises to the embedding space as contrastive views.

C.3 Implementation Details

We implement our RGCL with PyTorch [28] framework. For fair comparison, all models are initialized with the Xavier method [9] and optimized by the Adam optimizer [21]. All hyper-parameters of baseline models are searched following suggestions from the original papers. The batch size and embedding dimension are fixed to 4,096 and 64, respectively. The learning rate is searched from $\{0.0005, 0.001, 0.005, 0.01, 0.05\}$. The layer number of graph neural network is searched from $\{1, 2, 3\}$. We set $\mu = 0.1$ in Equation (15). The loss weight α is tuned from $\{1e-5, 5e-5, \dots, 1e-2\}$. The initial

hyper-parameter used for perturbation magnitude is chosen from $\{0.005, 0.01, \cdots, 1.0\}$. The search range of temperature coefficient τ is $\{0.05, 0.1, 0.2, 0.5, 1.0, 5.0, 10.0\}$. Early stopping is utilized as the convergence criterion. Specifically, we evaluate the performance on the validation dataset for each epoch, and stop the training process once there is no accuracy improvement for 10 consecutive epochs.

Algorithm 1 Learning Algorithm of RGCL

Input: User-item bipartite graph $\mathcal{G} = \{\mathcal{V}, \mathbf{A}\}$, adversarial loss weight μ, contrastive loss weight α, initialized perturbation magnitude ϵ, temperature coefficient τ, layer number K, batch size B, learning rate lr;

Parameter: Learnable parameters $\theta = \mathbf{E}$,

Output: RGCL Model;

1: **while** Model Not Convergence **do**
   // Calculate the decision boundary-aware perturbation
2:      Calculate the perturbation $\Delta_u^{(k)}, \Delta_i^{(k)}$ using Eq. (6);
   // Calculate the contrastive loss
3:      Generate perturbation-constrained random views $\mathbf{z}_u', \mathbf{z}_u''$, $\mathbf{z}_i', \mathbf{z}_i''$ using Eq. (6) and (7);
4:      Generate relation-aware adversarial-contrastive views $\mathbf{z}_u^{ac}, \mathbf{z}_i^{ac}$ using Eq. (10) and (11);
5:      Calculate multi-view contrastive loss $\mathcal{L}_{CL}$ using Eq. (12);
   // Calculate the adversarial loss
6:      Generate adversarial examples $\mathbf{z}_u^{adv}$ and $\mathbf{z}_i^{adv}$ using Eq. (13);
7:      Calculate adversarial loss $\mathcal{L}_{ADV}$ using Eq. (14);
   // Calculate the BPR loss;
8:      Calculate the BPR loss $\mathcal{L}_{BPR}$ using Eq. (1);
   // Model optimization
9:      Calculate total loss $\mathcal{L}$ using Eq. (15);
10:      Update model parameter θ using SGD;
11: **end while**
12: **return** θ;

C.4 Details on User and Item Grouping

In the following, we provide the specific details of partitioning the user and item groups in Experiment 5.4:

- **USER**: we split all users into five groups based on the number of user interaction while keeping the total number of each user group the same, which are denoted as $[G_0, G_1, G_2, G_3, G_4]$ in ascending order of interaction count.
- **ITEM**: we group all items based on their popularity into five groups and similarly, we keep the total number of each item group the same. Specifically, we adopt the decomposed Recall and NDCG metrics defined as follows:
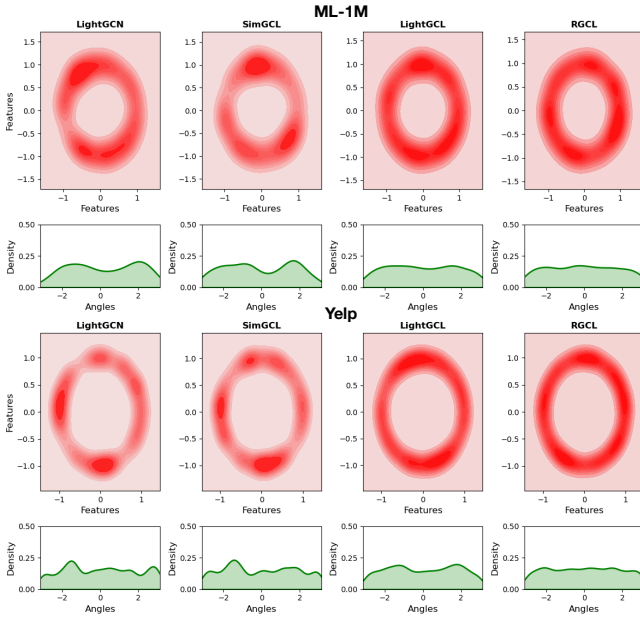
$$\text{Recall}(\mathbf{G}_i) = \frac{1}{M} \sum_{u \in \mathcal{U}} \frac{|\hat{l}_u \cap l_u^{\mathbf{G}_i}|}{|\hat{l}_u|},$$

$$\text{NDCG}(\mathbf{G}_i) = \frac{1}{M} \sum_{u \in \mathcal{U}} \frac{\sum_{j=1}^{|\hat{l}_u|} \mathbb{I}(\hat{l}_u(j) \in l_u^{\mathbf{G}_i})(\log_2(j+1))^{-1}}{\sum_{t=1}^{|\hat{l}_u|} (\log_2(t+1))^{-1}},$$

where $\hat{l}_u$ and l_u represent the predicted and real Top-N recommendation list of user u, respectively, and $\mathbb{I}(\cdot)$ is the indication

Table 4: Generalization evaluation on different GNN-based backbones.

Model	ML-1M						Yelp					
	R@10	N@10	R@20	N@20	R@50	N@50	R@10	N@10	R@20	N@20	R@50	N@50
GCMC	0.1676	0.2480	0.2526	0.2551	0.4073	0.2985	0.0520	0.0400	0.0867	0.0520	0.1623	0.0740
GCMC + RGCL	**0.1807**	**0.2608**	**0.2714**	**0.2707**	**0.4351**	**0.3176**	**0.0596**	**0.0463**	**0.0980**	**0.0596**	**0.1802**	**0.0835**
Improv.	+7.86%	+5.15%	+7.42%	+6.11%	+6.82%	+6.42%	+14.60%	+15.65%	+13.02%	+14.44%	+11.03%	+12.82%
NGCF	0.1763	0.2544	0.2673	0.2647	0.4297	0.3121	0.0506	0.0390	0.0842	0.0507	0.1570	0.0718
NGCF + RGCL	**0.1813**	**0.2565**	**0.2744**	**0.2683**	**0.4378**	**0.3165**	**0.0530**	**0.0405**	**0.0878**	**0.0526**	**0.1662**	**0.0752**
Improv.	+2.83%	+0.81%	+2.67%	+1.36%	+1.89%	+1.41%	+4.87%	+3.86%	+4.23%	+3.71%	+5.82%	+4.72%
GCCF	0.1753	0.2624	0.2611	0.2677	0.4171	0.3109	0.0512	0.0399	0.0851	0.0517	0.1582	0.0730
GCCF + RGCL	**0.1838**	**0.2679**	**0.2722**	**0.2747**	**0.4315**	**0.3195**	**0.0575**	**0.0451**	**0.0937**	**0.0576**	**0.1701**	**0.0798**
Improv.	+4.84%	+2.09%	+4.25%	+2.61%	+3.47%	+2.76%	+12.34%	+12.98%	+10.15%	+11.49%	+7.54%	+9.32%
LightGCN	0.1774	0.2581	0.2680	0.2670	0.4310	0.3137	0.0612	0.0479	0.1001	0.0614	0.1814	0.0850
LightGCN + RGCL	**0.1934**	**0.2694**	**0.2901**	**0.2821**	**0.4581**	**0.3321**	**0.0753**	**0.0591**	**0.1191**	**0.0744**	**0.2108**	**0.1010**
Improv.	+9.02%	+4.39%	+8.26%	+5.65%	+6.29%	+5.86%	+22.89%	+23.39%	+19.05%	+21.19%	+16.20%	+18.84%

Figure 7: Visualization of item representation and degree on ML-1M and Yelp datasets. Darker colors indicate more points falling within the region.

function. We use $l_u^{G_i}$ to denote the item recommendation list within the group G_i. Here, we set $|\hat{l}_u| = \min(|l_u|, K)$.

C.5 Learning Algorithm of RGCL

The overall learning algorithm of the proposed RGCL framework is summarized in Algorithm 1.

D More Experimental Analysis

D.1 Visualization of Representation (RQ4)

To better understand how RGCL promotes the uniformity of representations for preserving personalized node information, we visualize the learned item embeddings and user embeddings in Figure 7

and Figure 8, respectively. Specifically, we firstly map the learned node representations to 2-dimensional normalized vectors using t-SNE [31]. Then, we use Kernel Density Estimation (KDE) [2] to visualize the distribution of transformed feature representations. Moreover, for a clearer demonstration, we also visualize the density estimations of their angles, where angles are calculated using the function: $arctan2(y, x)$ for each instance (x, y). We can observe our RGCL shows a better uniform distribution on both users and items. This shows that RGCL can effectively learn high-quality representations by avoiding the bias caused by the dominance of advantaged users and items. Besides, correlating with the results in Table 1, RGCL achieves a win-win breakthrough in representation uniformization and performance improvement compared other baselines, suggesting the superiority of our designs.

D.2 Generalization Evaluation (RQ5)

To verify the generalization of our proposed model-agnostic framework, we employ RGCL framework on three other commonly used GNN-based backbones, i.e., GCMC [1], NGCF [33] and GCCF [4]. We summarize the experimental results in Table 4. From the table, we can see that RGCL generalizes well across different GNN-based backbones, further demonstrating the effectiveness and flexibility of our method. Additionally, the improvement based on the NGCF backbone is not significant, which we attribute to the redundant weight parameters and unnecessary nonlinear feature transformations of NGCF model, thus posing challenges to the model learning.

D.3 Case Study (RQ7)

In this section, we present a case study to intuitively show the effectiveness of our model to preserve the important semantic information of recommendation task. From the Figure 9, we can observe that user #315 prefers horror, action, and science fiction movies while showing less interest in comedy movies. Comparing the SimGCL and RGCL methods, although both original ranking results attain the correct ordering preferences for positive items and negative items, the introduction of noise perturbation for SimGCL
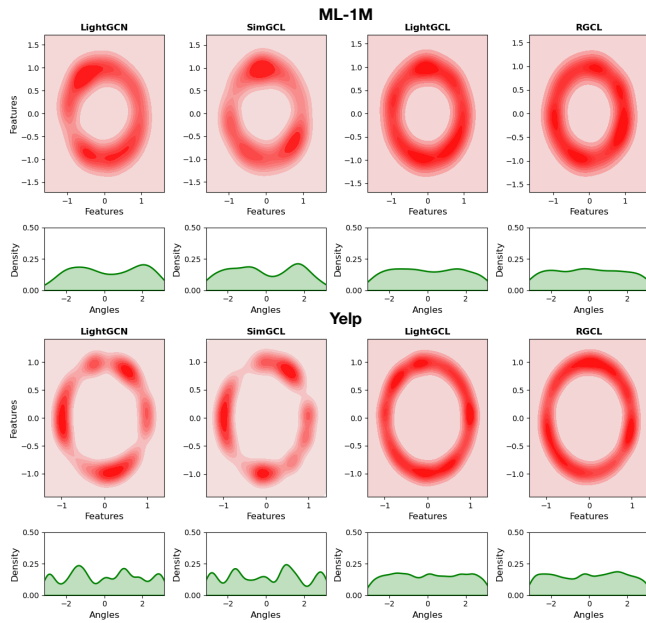
Figure 8: Visualization of user representation and degree on ML-1M and Yelp datasets. Darker colors indicate more points falling within the region.
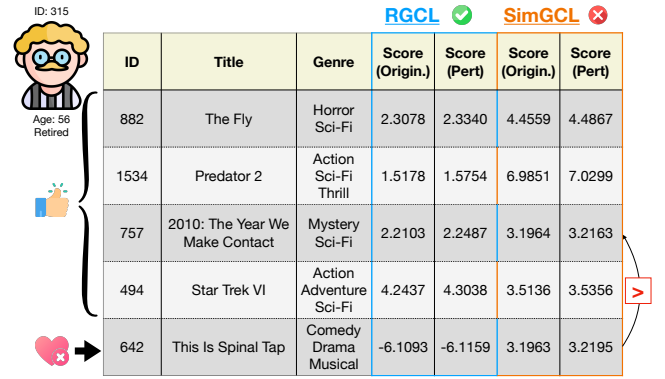

Figure 9: Case study on ML-1M dataset. The "Score (Origin.)" and "Score (Pert)" indicate predicted scores based on the original and contrastive augmented user and item embeddings, respectively. Best viewed in color.

baseline leads to a reversal in the predicted scores for movies #757 (liked movie) and movie #642 (disliked movie). It indicates that SimGCL baseline cannot reasonably control perturbations to preserve task-relevant information, resulting in irrational contrastive samples. In contrast, our proposed RGCL generates rational contrastive pairs and thus effectively improves model robustness and recommendation performance.