GCLCP: Graph Contrastive Learning with Convolutional Perturbation for Recommendation

Hao Pan[1], Lei Chen[1,2 (✉)], and Yangxun Ou[1]

[1] School of Computer Science and Technology, East China Normal University, Shanghai, China
[2] NPPA Key Laboratory of Publishing Integration Development, ECNUP, Shanghai, China,
`51265901004@stu.ecnu.edu.cn`, `lchen@cs.ecnu.edu.cn`

Abstract. Graph neural networks have been widely adopted in recommender systems for modeling user-item interactions as bipartite graphs. To address the inherent sparsity of these interactions, graph contrastive learning (GCL) enhances recommendation performance through self-supervised signals. However, existing GCL-based methods suffer from two key limitations: (i) graph perturbation-based contrastive views may distort structural information and degrade embedding quality; (ii) optimization via random negative sampling reduces training efficiency and recommendation quality. To address these challenges, we propose Graph Contrastive Learning with Convolutional Perturbation (GCLCP), a novel model for recommendation. GCLCP introduces perturbations to the neighborhood aggregation in graph convolution, generating contrastive views while preserving the graph structure. Furthermore, from the perspective of alignment and uniformity in representation learning, we incorporate these objectives into the GCL framework as loss functions, thereby improving both efficiency and accuracy without relying on negative sampling. Experiments on three real-world datasets demonstrate that GCLCP consistently outperforms baseline methods, achieving a 6.04% improvement in accuracy (NDCG@20) on the iFashion dataset. The code is available at https://github.com/pantiandu1/GCLCP.

Keywords: Recommendation · Contrastive Learning · Collaborative Filtering · Graph Neural Network

1 Introduction

Recommender systems are essential for personalizing user experiences and tackling information overload, making them a cornerstone of modern Internet services. Among them, collaborative filtering (CF) is the most widely adopted method, based on the intuitive principle that users with similar preferences are likely to make similar choices [1, 2]. With the advancements in graph neural networks [3], which have witnessed exponential growth in theoretical research and practical applications [4], graph collaborative filtering (GCF) has become a prominent paradigm in recommender systems, delivering remarkable performance and garnering significant attention in the field.

GCF models utilize an aggregation mechanism to capture multi-hop and indirect interactions, thereby improving recommendation accuracy [5]. However, their effectiveness is often limited by the inherent sparsity of user-item interactions. Graph contrastive learning (GCL) addresses this challenge through self-supervised learning, generating augmented views via edge dropping [6], or feature perturbation [7], and aligning them with contrastive loss functions such as InfoNCE to maximize mutual information [8]. These methods improve the robustness of representations under sparse supervision.

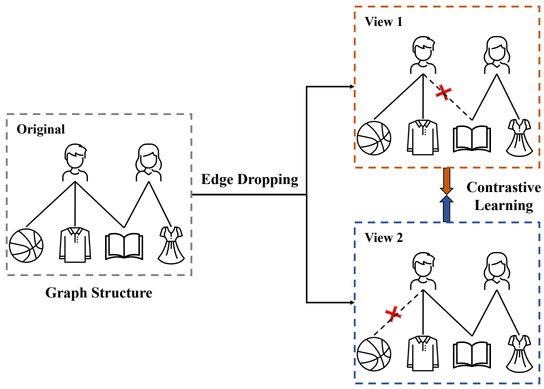

Fig. 1. An illustration of the misalignment issue in graph augmentation.

GCL models have shown potential in improving recommender systems, but they still face several challenges. A core assumption of GCL is that contrasting different graph augmentations can effectively capture essential information from user-item interaction graphs [6, 9]. However, such augmentations may distort the structural and semantic integrity of inherently sparse interaction graphs, leading to semantic misalignment. Taking Fig. 1 as an example, the original user-item interaction graph generates two views via random graph augmentations. However, in this case, random augmentation disrupts key interest signals within the user-item interaction graph. During contrastive learning, embeddings from distorted views are aligned, causing semantic misalignment where distinct user preferences, such as reading and sports, are incorrectly pulled closer. This misalignment of randomly augmented views can obscure critical signals about user preferences for specific items, thereby degrading recommendation performance.

Additionally, most GCL methods focus heavily on the contrastive learning objective while neglecting the roles of alignment and uniformity in representation learning [10]. Typically, they combine contrastive loss with Bayesian Personalized Ranking (BPR) loss, which relies on negative sampling [11]. The randomness of negative samples can slow down training convergence, while the inclusion of hard negatives—samples highly similar to positive ones—may result in degraded performance.

To address these challenges, we propose Graph Contrastive Learning with Convolutional Perturbation (GCLCP) for recommendation. First, we introduce a novel mechanism that generates contrastive learning views by applying perturbations to the neighborhood aggregation in graph convolution. This mechanism preserves the integrity of the original graph structure and essential information, enabling the model to learn more expressive and robust representations. Next, building on studies of alignment and uniformity in representation learning [11], we integrate these objectives into the GCL framework, directly optimizing the recommendation task without relying on negative sampling. This approach accelerates convergence, mitigates overfitting, and improves robustness. By combining these two strategies, our method achieves a more efficient training process and significantly enhances recommendation performance.

In summary, the key contributions of this work as follows:

- We propose a novel graph contrastive learning model for recommendation, named GCLCP, which generates contrastive views through convolutional perturbations on the interaction graph, thereby preserving structural integrity and learning more precise embedding representations.
- We integrate alignment and uniformity optimization into the GCLCP model, directly optimizing the recommendation task without negative sampling, thereby enhancing performance, training efficiency, and robustness.
- Extensive experiments on three real-world datasets demonstrate that our approach consistently outperforms baseline models, highlighting its effectiveness and superiority.

2 Related Work

2.1 Graph-based Collaborative Filtering

Collaborative filtering, which utilizes historical user-item interactions for recommendations, has long been central to recommender systems. With the rise of graph neural networks, user-item interactions are often modeled as bipartite graphs, where GNN-based methods aggregate high-order neighbor information to enhance representations, leading to the development of graph-based collaborative filtering. Representative models like NGCF [2], Multi-GCCF [12], and LightGCN [5] demonstrate the effectiveness of incorporating message passing, multi-relational graphs, and simplified graph convolutions, respectively. However, most research primarily focuses on encoder design, often overlooking other influential factors such as learning objectives and negative sampling strategies. Recent advances have explored alternative paradigms. For example, MixGCF [13] generates hard negatives by mixing positive samples with multi-hop neighbors instead of sampling from existing negatives. DirectAU [10] directly optimizes alignment and uniformity in the representation space. Although these approaches have shown promising results, they still suffer from challenges associated with data sparsity.

2.2 Graph Contrastive Learning for Recommendation

Recent studies have successfully integrated graph contrastive learning (GCL) into recommender systems to mitigate data sparsity. GCL provides self-supervised signals by creating augmented graph views and encouraging their representations to remain consistency. For example, SGL [6] applies graph augmentations, such as node dropout, edge dropout, and random walk, to generate contrastive views, which are optimized using the InfoNCE loss. SimGCL [7] challenges the necessity of graph augmentations and instead adds random noise into embeddings to generate contrastive views. NCL [14] and ProtoAU [15] enhances contrastive pair construction by introducing semantic prototypes. LightGCL [16] employs random SVD to reconstruct the adjacency matrix, preserving essential graph structure while enabling a low-rank optimization for contrastive learning. AU+ [17] introduces a zero-layer perturbation mechanism and directly optimizes alignment and uniformity objectives for recommendation.

3 Methodology

In this section, we introduce the proposed Graph Contrastive Learning with Convolutional Perturbation (GCLCP) method. GCLCP utilizes graph convolutional networks to generate user and item embeddings for both recommendation and contrastive learning tasks. To enable contrastive learning, GCLCP applies convolutional perturbations during the graph convolution process to construct two distinct views. Additionally, alignment and uniformity objectives are employed to evaluate and enhance the quality of learned representations. The overall framework is illustrated in Fig. 2.

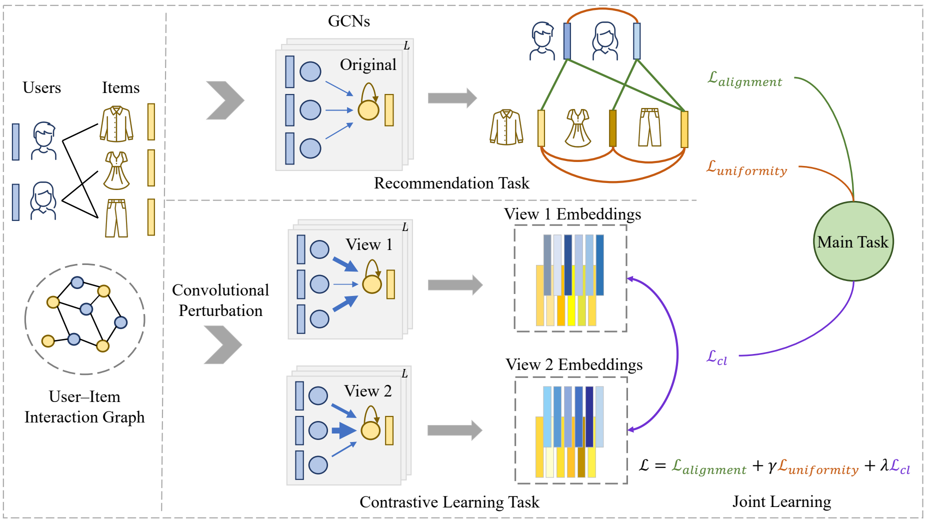

Fig. 2. The overall framework of the proposed GCLCP.

3.1 Graph Collaborative Filtering Backbone

To encode the interaction patterns between users and items, we follow the common graph collaborative filtering paradigm by embedding them into a d-dimensional latent space.

Let U and I denote the sets of users and items, respectively. The interaction matrix $A \in \{0,1\}^{|U|\times|I|}$ indicates the interaction relationships between users and items, where each entry $A_{u,i}$ is 1 if user u has interacted with item i, and 0 otherwise. Next, we generate embeddings $e_u \in R^d$ for user $u \in U$ and $e_i \in R^d$ for item $i \in I$.

Our method GCLCP adopts the widely used and effective graph encoder LightGCN [5] as its backbone, the neighbor information aggregation process as follows:

$$e_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(l)}, \quad e_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{(l)} \tag{1}$$

where $e_u^{(l)}$ and $e_i^{(l)}$ respectively denote the refined embedding of user u and item i after l layers propagation, $\mathcal{N}_u$ denotes the set of items that are interacted by user u, $\mathcal{N}_i$ denotes the set of users that interact with item i.

With the embeddings of nodes obtained from the L-th layer, the base model computes the representations of users and items as follows:

$$e_u = \frac{1}{L+1} \sum_{l=0}^{L} e_u^{(l)}, \quad e_i = \frac{1}{L+1} \sum_{l=0}^{L} e_i^{(l)} \tag{2}$$

To facilitate implementation and discussion, we also provide the matrix form of the propagation process as:

$$E = \frac{1}{L+1}\left(E^{(0)} + \tilde{A}E^{(0)} + \cdots + \tilde{A}^{L}E^{(0)}\right) \tag{3}$$

where $E^{(0)} \in R^{(|U|+|I|)\times d}$ is the randomly initialized embeddings of both users and items, $\tilde{A} \in R^{(|U|+|I|)\times(|U|+|I|)}$ is the normalized undirected adjacency matrix [5].

Finally, the predicted preference score, obtained as the inner product of the user and item embeddings, can be expressed as:

$$\hat{y}_{u,i} = {e_u}^{\top} e_i \tag{4}$$

3.2 Alignment and Uniformity Objectives

Traditional GCL-based recommendation methods commonly utilize on the Bayesian personalized ranking (BPR) loss [11] to optimize item ranking. Formally, the objective function of BPR loss is defined as:

$$\mathcal{L}_{BPR} = \frac{1}{|A|} \sum_{(u,i)\in A} -log[sigmoid(e_u^{\top} e_i - e_u^{\top} e_{i^-})] \tag{5}$$

where i^- represents a randomly sampled negative item that the user has not interacted with. The loss function is designed to maximize the probability that the target item receives a higher score than the sampled items.

We observe that the BPR loss is sensitive to negative sampling. When the sampled item is not truly irrelevant to the user, it may introduce noisy or misleading supervision, making it difficult for the model to learn effective embeddings. This issue slows convergence and degrades the model's overall performance.

Inspired by recent studies [10] in representation learning, we address these limitations by incorporating alignment and uniformity into the optimization process. Specifically, given the distributions of users $p_{user}(\cdot)$, items $p_{item}(\cdot)$, and positive pairs $p_{pos}(\cdot,\cdot)$, the alignment and uniformity losses are defined as:

$$\mathcal{L}_{alignment} = \mathbb{E}_{(u,i)\sim p_{pos}}\|z_u - z_i\|_2^2$$

$$\mathcal{L}_{uniformity} = \frac{1}{2}log\mathbb{E}_{(u,u')\sim p_{user}}e^{-2\left\|z_u - z_{u'}\right\|_2^2} + \frac{1}{2}log\mathbb{E}_{(i,i')\sim p_{item}}e^{-2\left\|z_i - z_{i'}\right\|_2^2} \quad (6)$$

where z represents the L_2-normalized embedding of its corresponding embedding e. The alignment loss pulls embeddings of users and items with historical interactions closer together in the representation space. Concurrently, the uniformity loss encourages a more evenly distributed embedding space.

By jointly optimizing alignment and uniformity, the model balances local consistency and global discrimination. Alignment ensures that user preferences are accurately captured by drawing interacted pairs closer, while uniformity prevents representation collapse and encourages diversity across all users and items. This complementary effect leads to more accurate and reliable recommendation performance.

3.3 Contrastive Learning with Convolutional Perturbation

Following existing GCL-based recommendation approaches, we adopt the InfoNCE loss to capture the invariance between two augmented views, thereby enhancing the quality of the learned embeddings and improving overall recommendation performance. The contrastive loss is typically defined as:

$$\mathcal{L}_{cl} = \sum_{i\in\mathcal{B}} -log\frac{exp\left(e_i'^{\top}e_i''/\tau\right)}{\sum_{j\in\mathcal{B}}exp\left(e_i'^{\top}e_j''/\tau\right)} \quad (7)$$

where i and j represent users/items in a sampled batch $\mathcal{B}$, e' and e'' denote the d-dimensional node representations learned from two different graph augmentation views, and τ is the temperature coefficient. The CL loss encourages consistency between e' and e'', which are the augmented representations of the same node i and serve as positive pairs, while minimizing the agreement between e_i' and e_j'', which act as negative pairs.

We observe that some GCL methods generate contrastive views through graph perturbations [6], which can often disrupt the semantic structure of the graph. To obtain effective contrastive views without compromising the structural integrity, we focus on the neighborhood aggregation phase in graph convolution. Specifically, we introduce convolutional perturbation to adjust the influence of neighboring nodes during aggregation, generating views tailored for contrastive learning.

Formally, given a node i and its representation e_i in the d-dimensional embedding space, we define the following convolutional perturbation-based augmentation:

$$e_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{\Delta_{u,i}}{\sqrt{|\mathcal{N}_u|}\sqrt{|\mathcal{N}_i|}} e_i^{(l)}, \quad e_i^{(l+1)} = \sum_{i \in \mathcal{N}_i} \frac{\Delta_{i,u}}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_u|}} e_u^{(l)} \tag{8}$$

where the noise term $\Delta_{u,i}$ is sampled from a uniform distribution $U(1-\epsilon, 1+\epsilon)$, where $\epsilon > 0$ represents the convolutional perturbation magnitude, and $\Delta_{u,i} = \Delta_{i,u}$. The first constraint bounds the perturbation magnitude, preventing excessive deviation from the original embedding and retaining its semantic information. The second constraint enforces symmetry, maintaining the graph's topological structure, which is essential for the performance of the GNN model. Together, these constraints simultaneously preserving the integrity of the graph structure while enhancing the robustness of the model.

In the implementation, we apply convolutional perturbation to the normalized adjacency matrix $\tilde{A}$, thereby affecting the global information propagation process. The perturbation is represented by a noise matrix $\Delta \in R^{(|U|+|I|)\times d}$, where each entry $\Delta_{i,j}$ satisfies the previously defined constraints. The final node representations are computed as follows:

$$E = \frac{1}{L+1}\left(E^{(0)} + \left(\tilde{A} \circ \Delta\right)E^{(0)} + \cdots + \left(\tilde{A} \circ \Delta\right)^L E^{(0)}\right) \tag{9}$$

By applying convolutional perturbation to the neighborhood aggregation process, we obtain two views E', E'', which are then used to compute the contrastive learning loss via Eq. 7, serving as auxiliary supervision for the recommendation task.

3.4 Model Training

We leverage a multi-task training strategy to jointly optimize the recommendation task (Eq. 6) and the contrastive learning task (Eq. 7). The overall loss function defined as:

$$\mathcal{L} = \mathcal{L}_{alignment} + \gamma\mathcal{L}_{uniformity} + \lambda\mathcal{L}_{cl} + \lambda_r\|\Theta\|_2^2 \tag{10}$$

where Θ is the set of model parameters, specifically the user and item embeddings. The hyperparameters γ and λ control the weights of $\mathcal{L}_{uniformity}$ and $\mathcal{L}_{cl}$, respectively. Additionally, λ_r controls the weight of the L_2 regularization loss.

3.5 Time Complexity Analysis

In this part, we analyze the time complexity of our proposed model GCLCP. Let $|A|$ represent the number of edges in the graph, d be the embedding size, L denote the number of GCN layers, and B denote the batch size. The detailed analysis is as follows:

First, adjacency matrix construction involves generating two perturbed views in addition to the original graph, yielding a complexity of $O(6|A|)$. Second, the graph convolution is performed by three parallel encoders, resulting in a total complexity of $O(6|A|Ld)$. Third, for the alignment and uniformity losses, each batch requires $O(Bd)$ and $O(2B^2d)$ time respectively, leading to an overall complexity of $O\big(|A|d(1+2B)\big)$

over all batches. Finally, the contrastive learning loss, which computes pairwise similarities within each batch, has a complexity of $\mathcal{O}\big(|A|d(2+2B)\big)$.

Compared with other GCL-based approaches, our contrastive loss maintains the same time complexity. Although the complexity of the recommendation task has increased from $\mathcal{O}(2Bd)$ to $\mathcal{O}\big(|A|d(1+2B)\big)$, the overall time complexity remains in the same order. As will be shown in Section 4.3, GCLCP consistently achieves competitive recommendation performance within just 10 epochs, and its overall training time is often shorter than other methods.

4 Experiments

In this section, we first introduce the experimental settings. Then we conduct a series of experiments to evaluate the performance of the proposed GCLCP on the recommendation task. Additionally, we compare training dynamics and efficiency, perform an ablation study, and analyze the sensitivity to hyperparameters.

4.1 Experimental Settings

Datasets. To evaluate the performance of the proposed GCLCP model, we conduct experiments on three benchmark datasets: iFashion [18], Amazon-Books [19], and Yelp2018 [5]. These datasets vary in domain, scale, and data density. The detailed statistics of the datasets are summarized in **Table 1.** For each dataset, we randomly partition the data into training, and testing sets with a proportion of 8:2, ensuring comprehensive model evaluation.

Table 1. Basic information of experimental datasets.

Datasets	#Users	#Items	#Interactions	Density
iFashion	300,000	81,614	1,607,813	0.00007
Amazon-Book	52,463	91,599	2,984,108	0.00062
Yelp2018	31,668	38,048	1,561,406	0.00130

Baselines. We compare our method GCLCP with the following baseline methods:

- **LightGCN** [5]: This method simplifies graph convolutional networks by focusing on the linear propagation of collaborative signals, enhancing both efficiency and recommendation performance.
- **DirectAU** [10]: This method revisits representation learning by directly optimizing alignment and uniformity on the hypersphere, replacing the traditional BPR loss.
- **MixGCF** [13]: This method proposes a hop-mixing technique to generate hard negatives by interpolating embeddings for graph collaborative filtering.
- **NCL** [14]: This method introduces a prototypical contrastive framework that jointly captures structural and semantic user-item correlations to improve performance.

- **SGL** [6]: This method leverages graph contrastive learning with edge dropout to improve recommendation performance.
- **SimGCL** [7]: This method introduces random noise to the embedding space to generate contrastive views, simplifying the augmentation process while improving recommendation performance and training efficiency.
- **LightGCL** [16]: This method constructs contrastive views through random singular value decomposition.
- **AU⁺** [17]: This method introduces a zero-layer perturbation mechanism and directly optimizes alignment and uniformity objectives for recommendation.

Evaluation Metrics. To measure the performance of our recommendation model, we adopt widely used evaluation metrics: Recall@20 and NDCG@20.

Model Parameters. To ensure a fair comparison, we adopt the optimal hyperparameter settings reported in the original papers of the baseline models. The embedding size is set to 64, and the batch size is fixed at 2048. All models are optimized using the Adam optimizer with a learning rate of 0.001 and an L_2 regularization coefficient of 0.0001. For GNN-based methods, the number of graph convolution layers is set to 3. For all GCL-based models, we empirically set the default temperature $\tau = 0.2$, as it performs best in most cases.

4.2 Overall Performance

Table 2. Performance comparison on three datasets among different models.

Method	iFashion		Amazon-Book		Yelp2018	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
LightGCN	0.0896	0.0416	0.0411	0.0315	0.0639	0.0525
DirectAU	0.1086	0.0506	0.0477	0.0383	0.0692	0.0575
MixGCF	0.1075	0.0516	0.0485	0.0378	0.0691	0.0577
NCL	0.0914	0.0428	0.0442	0.0343	0.0666	0.0555
SGL	0.1107	0.0528	0.0478	0.0379	0.0675	0.0555
SimGCL	0.1143	0.0544	0.0515	0.0414	0.0721	0.0601
LightGCL	0.0956	0.0443	0.0497	0.0390	0.0684	0.0561
AU⁺	<u>0.1157</u>	<u>0.0546</u>	<u>0.0531</u>	<u>0.0430</u>	<u>0.0728</u>	<u>0.0612</u>
GCLCP	**0.1207**	**0.0579**	**0.0544**	**0.0445**	**0.0746**	**0.0620**
Improv.	4.32%	6.04%	2.45%	3.49%	2.47%	1.31%

Table 2 presents the performance of various baseline CF methods and our proposed GCLCP model across three datasets. Based on the experimental results, we derive the following key observations:

- Adding contrastive learning as an auxiliary task consistently improves the performance of LightGCN. GCL-based methods outperform the backbone, demonstrating the effectiveness of contrastive learning in alleviating data sparsity. Among them, SimGCL and AU$^+$, which add random noise to embeddings, outperforms graph-perturbation-based methods like SGL by better preserving structural information. In contrast, our method perturbs the convolutional process itself, effectively leveraging the graph's propagation structure and achieving superior performance.
- Compared to LightGCN, methods like DirectAU and MixGCF achieve improvements by optimizing the negative sampling process within the BPR loss. Among these, DirectAU achieves the best results.
- Our proposed GCLCP model achieves the best performance. It outperforms the strongest baseline AU$^+$, with improvements of 4.32% in Recall@20 and 6.04% in NDCG@20 on the iFashion dataset. While both GCLCP and AU$^+$ preserve the original graph structure, GCLCP introduces convolutional perturbations to better exploit the message-passing nature of GNNs, resulting in more effective contrastive views. Compared to graph-perturbation-based methods like SGL, GCLCP maintains structural integrity, further emphasizing the importance of preserving graph structure in contrastive learning. Moreover, Our method GCLCP performs particularly well on the sparsest dataset, iFashion, indicating its superior ability to exploit the underlying graph structure.

4.3 Training Dynamics and Efficiency Analysis

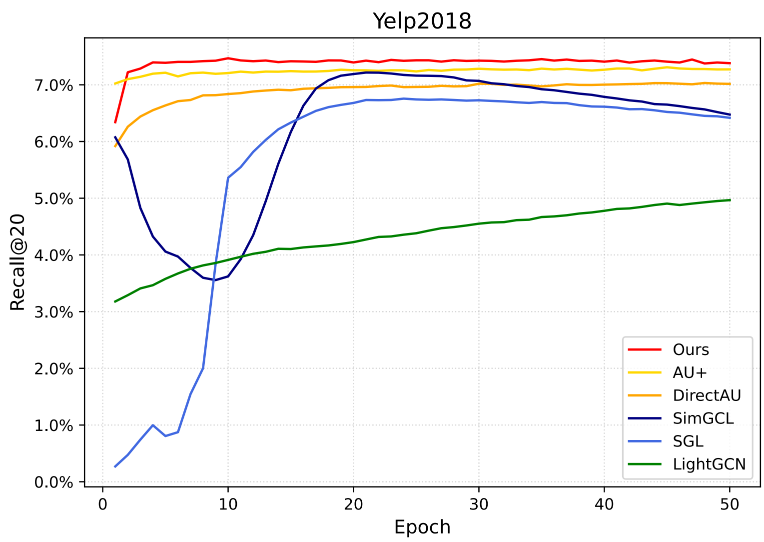

Fig. 3. The performance curve in the first 50 epochs.

In this part, we analyze the training dynamics and training efficiency of GCLCP and representative baseline methods on the Yelp2018 dataset.

As shown in Fig. 3, GCLCP, DirectAU, and AU^+ exhibit similar learning curves, maintaining strong and stable performance in the later stages of training. This can be attributed to their shared emphasis on optimizing alignment and uniformity, which facilitates faster convergence. Unlike other GCL methods that typically require over 20 epochs, GCLCP and AU^+ achieve strong early-stage performance and further improve through contrastive learning. Moreover, unlike AU^+ and SimGCL, GCLCP introduces contrastive views via convolutional perturbation, leading to more effective exploitation of the graph's message-passing structure. The superior performance of GCLCP over AU^+ suggests that perturbing the convolutional process is better suited for alignment and uniformity optimization.

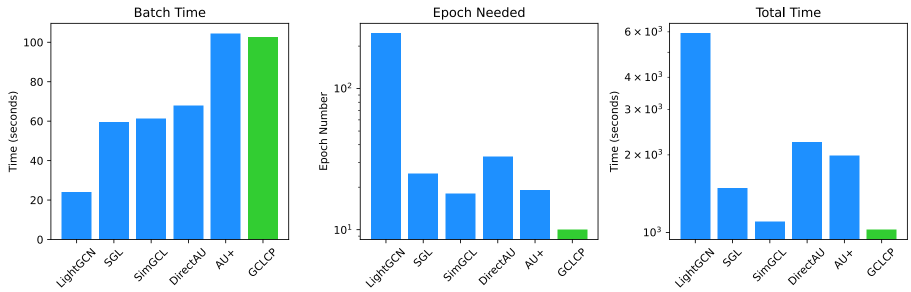

Fig. 4. The training speed of compared methods.

As illustrated in Fig. 4, We compare several aspects of training efficiency across different methods, including batch time, the number of epochs to convergence, and total training time. GCLCP incorporates contrastive learning and alignment–uniformity losses on top of LightGCN, which lead to a slightly longer batch time. Nevertheless, this overhead is offset by the significantly reduced number of training epochs needed to reach convergence. As a result, GCLCP achieves competitive performance with the shortest overall training time among all methods.

Combining the training efficiency analysis with training dynamics, we observe that GCLCP achieves strong and stable recommendation performance within a short training time, demonstrating its high efficiency.

4.4 Ablation Study

In this part, we evaluate the contribution of each component in GCLCP through an ablation study. Specifically, we design two variants: one without contrastive learning task (referred to as *GCLCP-w/o-CL*), and the other replacing the alignment and uniformity objectives with the BPR loss (referred to as *GCLCP-w/o-AU*).

Moreover, we also conduct experimental discussions on different forms of convolutional perturbations. $GCLCP_a$ represents asymmetric perturbation which not satisfying symmetric perturbation constraint $\Delta_{u,i} = \Delta_{i,u}$, and $GCLCP_g$ represents Gaussian distribution noise perturbation, respectively. The results are presented in Table 3.

Table 3. The ablation study on GCLCP and its variants.

Method	Amazon-Book		Yelp2018	
	Recall	NDCG	Recall	NDCG
LightGCN	0.0411	0.0315	0.0639	0.0525
GCLCP-w/o-CL	0.0477	0.0383	0.0692	0.0575
GCLCP-w/o-AU	0.0512	0.0410	0.0720	0.0591
$GCLCP_a$	0.0539	0.0441	0.0738	0.0615
$GCLCP_g$	0.0533	0.0438	0.0736	0.0613
GCLCP	**0.0544**	**0.0445**	**0.0746**	**0.0620**

From the table, we observe that removing or replacing either module leads to a degradation in performance: (i) *GCLCP-w/o-CL* optimizes only the alignment and uniformity objectives, relying on labeled data for representation learning. This limits the generalizability of the learned representations and results in inferior recommendation performance. (ii) *GCLCP-w/o-AU*, which adopts BPR loss for optimization, suffers from slower convergence due to its dependence on negative sampling and insufficient uniformity, ultimately yielding suboptimal performance. By integrating contrastive learning with alignment and uniformity objectives, GCLCP achieves both efficient training and strong performance.

As for the two variants with different convolutional perturbation strategies, $GCLCP_a$ and $GCLCP_g$, both exhibit a slight performance decline. Among them, $GCLCP_g$ shows a more notable decline, whereas $GCLCP_a$ retains relatively strong performance. We argue that this difference is attributed to the lower controllability of Gaussian noise, which complicates the regulation of perturbation magnitudes. In contrast, asymmetric perturbations may better adapt to specific datasets. These findings highlight the necessity of constraining noise distribution and symmetry to ensure stable performance and improve model generalization.

4.5 Hyperparameter Sensitivity Analysis

In this part, we conduct a hyperparameter sensitivity analysis on the Yelp2018 dataset, with the results presented in Fig. 5.

- Impact of λ: We fix ϵ at 0.1 and γ at 0.5, and then vary λ across a set of representative values to investigate the influence of contrastive learning loss magnitude. The optimal λ value for GCLCP is 0.5 on the Yelp2018 dataset. We argue that an excessively high λ value leads the model to overprioritize distinguishing similar items, which undermines recommendation performance.
- Impact of ϵ: We fix λ at the optimal value and adjust ϵ to examine the impact of convolutional perturbation magnitude on performance. The optimal ϵ value for GCLCP is 0.05, the result show that lower ϵ value leads to better performance, as excessive perturbation reduces the alignment between embeddings from the two views, thereby diminishing the effectiveness of the contrastive learning task.

– Impact of γ: We fix λ and ϵ at their optimal values, then adjust the weight of uniformity loss γ. It is found that the best performance is achieved when γ is set to 0.5. This indicates that effective recommendation requires the learned embeddings to strike a balance between alignment and uniformity losses, as both are critical.

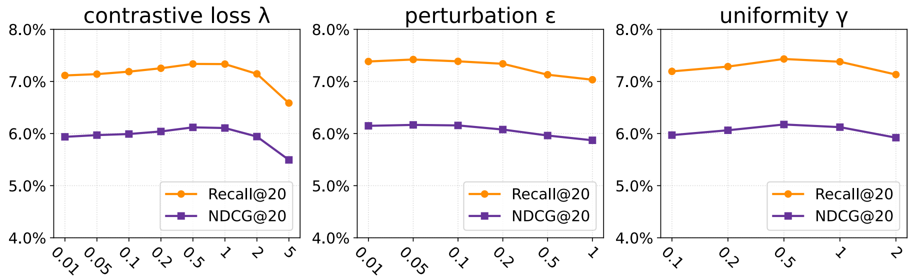

Fig. 5. Influence of the magnitude of hyperparameters on Yelp2018.

5 Conclusion

In this paper, we investigate graph contrastive learning for recommendation, identify the issues of view misalignment in graph augmentation and the inefficiency caused by random negative sampling during optimization. To address these problems, we propose the Graph Contrastive Learning with Convolutional Perturbation (GCLCP) method, which introduces convolutional perturbations to generate contrastive views. This approach preserves the structural information of the graph and enhances recommendation performance. Furthermore, we incorporate alignment and uniformity as objectives for the recommendation task, achieving improvements in both effectiveness and efficiency. Comprehensive experiments demonstrate that the proposed GCLCP method outperforms other GCL-based methods, and significantly improves training efficiency.

References

1. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The adaptive web: methods and strategies of web personalization, pp. 291–324. Springer (2007)
2. Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval. pp. 165–174 (2019)
3. Li, Z.P., Wang, S.G., Zhang, Q.H., Pan, Y.J., Xiao, N.A., Guo, J.Y., Yuan, C.A., Liu, W.J., Huang, D.S.: Graph pooling for graph-level representation learning: a survey. Artificial Intelligence Review 58(2), 45 (2024)
4. Li, Z.P., Su, H.L., Zhu, X.B., Gribova, V., Filaretov, V.F., Huang, D.S.: SSPool: a simple siamese framework for graph infomax pooling. IEEE Transactions on Network Science and Engineering 11(1), 463–470 (2023)

5. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: 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. pp. 639–648 (2020)
6. Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X.: Self-supervised graph learning for recommendation. In: Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. pp. 726–735 (2021)
7. Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., Nguyen, Q.V.H.: 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. pp. 1294–1303 (2022)
8. Bachman, P., Hjelm, R.D., Buchwalter, W.: Learning representations by maximizing mutual information across views. Advances in neural information processing systems 32 (2019)
9. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the web conference 2021. pp. 2069–2080 (2021)
10. Wang, C., Yu, Y., Ma, W., Zhang, M., Chen, C., Liu, Y., Ma, S.: Towards representation alignment and uniformity in collaborative filtering. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. pp. 1816–1825 (2022)
11. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618 (2012)
12. Sun, J., Zhang, Y., Ma, C., Coates, M., Guo, H., Tang, R., He, X.: Multi-graph convolution collaborative filtering. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 1306–1311. IEEE (2019)
13. Huang, T., Dong, Y., Ding, M., Yang, Z., Feng, W., Wang, X., Tang, J.: 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. pp. 665–674 (2021)
14. Lin, Z., Tian, C., Hou, Y., Zhao, W.X.: Improving graph collaborative filtering with neighborhood-enriched contrastive learning. In: Proceedings of the ACM web conference 2022. pp. 2320–2329 (2022)
15. Ou, Y., Chen, L., Pan, F., Wu, Y.: Prototypical contrastive learning through alignment and uniformity for recommendation. In: 2024 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2024)
16. Cai, X., Huang, C., Xia, L., Ren, X.: LightGCL: Simple yet effective graph contrastive learning for recommendation. In: The Eleventh International Conference on Learning Representations (2023)
17. Ouyang, Z., Zhang, C., Hou, S., Zhang, C., Ye, Y.: How to improve representation alignment and uniformity in graph-based collaborative filtering? In: Proceedings of the International AAAI Conference on Web and Social Media. vol. 18, pp. 1148–1159 (2024)
18. Chen, W., Huang, P., Xu, J., Guo, X., Guo, C., Sun, F., Li, C., Pfadler, A., Zhao, H., Zhao, B.: Pog: personalized outfit generation for fashion recommendation at alibaba ifashion. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 2662–2670 (2019)
19. McAuley, J., Targett, C., Shi, Q., Van Den Hengel, A.: Image-based recommendations on styles and substitutes. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp. 43–52 (2015)