Supplementary Materials

1 Model complexity analysis

Specifically, we use $|V|$ to denote the total number of nodes in the user-item interaction graph G and $|E|$ to denote the total number of edges in G. B is the batch size, d is the embedding dimension, and L is the number of layers. In our model, the process for the adjacency matrix involves initialization and transformation and considers the number of non-zero elements, which gives a time complexity of $O(2|E|)$. For graph convolution, based on LightGCN as the encoder L-layer complexity is $O(L|E|d)$, and due to the application of the siamese network structure, then each batch is executed twice, so the total time complexity of graph convolution is $O(2L|E|d)$. In addition, the BPR loss computation involves the user-item interaction pairs in the batch with a complexity of $O(2Bd)$, which in our siamese structure needs to be computed twice, hence $O(4Bd)$. For the computation of the self-supervised loss InfoNCE, we use embedding augmentation, hence $O(4B|V|d)$ complexity. Finally, the alignment loss, which improves alignment, mainly involves the normalization of the embedding vectors and the computation of the differences, with a total complexity of $O(2Bd)$.

2 Datasets

- **ml-1M**[1]: A widely used movie recommendation dataset, collected by the GroupLens research team from their movie website. It includes 1,000,209 ratings generated by 6,040 users on 3,900 movies. Each rating data contains the user ID, movie ID, rating (on a scale of 1-5), and timestamp (representing the time when the rating was made).
- **LastFM**[2]: It originates from the Last.fm website, which provides music social networking services, and has been widely used in music recommendation research. The dataset records user information, song and artist details, and user listening history. In our study, we mainly focus on the interaction between users and artists to recommend artists to users. The dataset used in our study contains 92,834 user-artist interactions involving 1,892 users and 17,632 artists.
- **Yelp2018**[3]: It originates from the Yelp Challenge and contains a large number of active interactions between many users and business entities. It has become a widely adopted benchmark for evaluating the performance of various models in RS. The dataset we use includes 31,668 users, 38,048 business entities, and a total of 1,561,406 interactions between them.
- **Gowalla**[4]: This dataset comes from a location-based social network platform where users share their locations through check-ins. It consists of 29,859 users, 40,989 items, and 1,027,464 interactions.
- **Alibaba-iFashion**[5]: This dataset from Alibaba's e-commerce platform focuses on fashion clothing. It is a particularly sparse dataset in the RS, with a density of only 0.00007. It involves 300,000 users, 81,614 items, and a total of 1,607,813 interactions.

[1] http://files.grouplens.org/datasets/movielens/
[2] http://ir.ii.uam.es/hetrec2011
[3] https://www.yelp.com/dataset
[4] https://snap.stanford.edu/data/loc-gowalla.html
[5] https://www.alibabagroup.com/

3 Baselines

- **BPR-MF** [5]: A general recommendation moel to solve the sorting problem in RS. It selects and arranges items based on the user's behavioral data to ensure the front row items are more likely to appeal to the user. This algorithm is based on the largest posterior estimation, derived from Bayesian analysis.
- **NCF** [2]: A neural collaborative filtering model applies neural networks to model implicit feedback. Differing from matrix factorization, NCF replaces inner products with neural networks and utilizes multi-layer perceptrons to learn user-item interactions. This approach enhances the nonlinear capabilities of the modeling process.
- **NGCF** [7]: A recommendation model that employs GNN to directly model the user-item interaction graph. This approach is capable of capturing high-order connections between users and items, thereby enhancing the accuracy and relevance of the recommendations.
- **LightGCN** [1]: A simplified and enhanced recommendation model based on Graph Convolution Network (GCN) retains the core neighbor aggregation operation and linearly propagates user and item embedding vectors on the user-item interaction graph.
- **SGL** [8]: A GCL-based recommendation model employs three augmentation versions — node dropout, edge dropout, and random walk — to generate multiple enhanced views of a node. By bringing the consistency closer between different views of the same node, while simultaneously pushing the consistency further between views of different nodes, it guides the representation learning of nodes.
- **SimpleX** [4]: A simple baseline based on collaborative filtering emphasizes that the performance of collaborative filtering-based recommendation models depends not only on the encoder but also on the loss function and negative sampling. To address this, the approach proposes the use of cosine contrastive loss in current recommendation models.
- **NCL** [3]: A neighborhood-enhanced contrastive learning method is introduced that leverages neighbors within the graph structure and semantic space to mitigate data sparsity in graph collaborative filtering.
- **XSimGCL** [9]: An extremely simple graph contrastive learning method is introduced, which does not rely on complex graph augmentations, but instead uses simple random noise embedding augmentations.
- **DirectAU** [6]: A novel loss function is introduced, designed to quantify alignment and uniformity from the perspectives of representation alignment and uniformity. This approach aims to significantly improve recommendation performance.

4 Evaluation metrics

Hit@K: measures the consistency between model predictions and actual user interactions in the Top-K recommendation list. It calculates the ratio of the number of hits for each user in the Top-K recommendations to their total number of actual interactions. This ratio reflects the effectiveness of the recommendation system in identifying users' true interests in the top portion of the recommendation list. Its equation is shown in Eq. (1), where $HitsCount@K_{user}$ is the number of hits for each user within the Top-K recommended items, N_{user} represents the total number of interaction items for each user in the test set.

$$\text{Hit@K} = \frac{\sum_{user} \text{HitsCount@K}_{user}}{\sum_{user} N_{user}} \tag{1}$$

Precision@K: measures the accuracy of a recommendation system in predicting items of interest for users within the first K items of the recommendation list. It calculates the ratio of the total number of hits across all users to the total number of recommended items, which is K for each user. This ratio reflects the overall accuracy of the recommendation list. It is defined as shown in Eq. (2). Where $|U|$ is the total number of users.

$$\text{Precision@K} = \frac{\sum_{user} \text{HitsCount@K}_{user}}{|U| \times K} \tag{2}$$

Recall@K: which is similar to a binary classification metric, is used to measure how many of the items that the user is actually interested in have been successfully appeared in the Top-K positions of the recommendation list given by the model. Its equation is shown in Eq. (3). Where $R(u,K)$ is the set of Top-K recommended items for user u, and I_u is the set of items that user u has actually interacted with.

$$Recall@K = \frac{1}{|U|} \sum_{u \in U} \frac{|\{i \in R(u,K) \cap I_u\}|}{|I_u|} \tag{3}$$

NDCG@K: a measure of the ranking quality of recommendation lists in RS combines the effects of item relevance and ranking position. Its equation is shown in Eq. (4). Where rel_i is the true relevance score of the i-th recommended item. And $|REL|$ denotes the number of items in the set composed of the Top-K recommendations sorted by true relevance in descending order.

$$NDCG@K = \frac{DCG@K}{IDCG@K} = \frac{\sum\limits_{i=1}^{K} \frac{rel_i}{log_2(i+1)}}{\sum\limits_{i=1}^{|REL|} \frac{rel_i}{log_2(i+1)}} \tag{4}$$

5 Alleviate over-smoothing performance

It is well-known that GNNs are susceptible to over-smoothing effects, which lead to similar feature representations among different nodes and thus diminish the model's ability to distinguish between them. To investigate the specific impact of increasing the number of layers on model performance, we conducted a comparative analysis between the GNN-based SiamGCL and SGL. As illustrated in Figure 1, taking Yelp2018 as an example, we observed that SGL exhibits improved performance in Recall@10, Recall@20, and Recall@50 metrics with an increase in layers, although this improvement gradually diminishes at higher layer counts. In contrast, SiamGCL demonstrated more balanced and consistent performance enhancements across all layers, particularly maintaining high recommendation accuracy even at deeper levels.

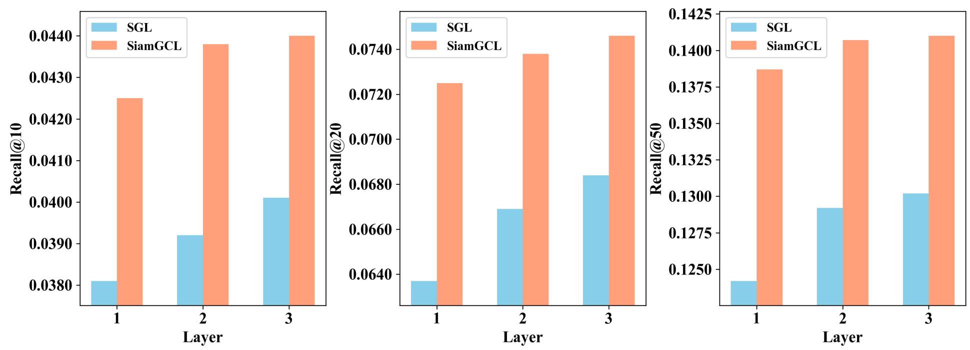
References

[1] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 639–648. https://doi.org/10.1145/3397271.3401063

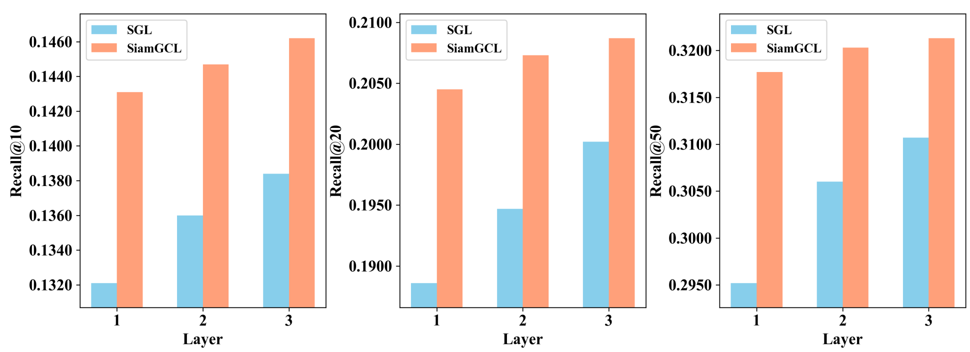
[2] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 173–182. https://doi.org/10.1145/3038912.3052569

[3] Zihan Lin, Changxin Tian, Yupeng Hou, and Wayne Xin Zhao. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, Frédérique Laforest, Raphaël Troncy, Elena Simperl, Deepak Agarwal, Aristides Gionis, Ivan Herman, and Lionel Médini (Eds.). ACM, 2320–2329. https://doi.org/10.1145/3485447.3512104

[4] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A Simple and Strong Baseline for Collaborative Filtering. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland,*



(a) Yelp2018



(b) Gowalla

Fig. 1. Performance comparison of different layers on Yelp2018 and Gowalla.

Australia, November 1 - 5, 2021, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 1243–1252. https://doi.org/10.1145/3459637.3482297

[5] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461.

[6] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards Representation Alignment and Uniformity in Collaborative Filtering. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 1816–1825. https://doi.org/10.1145/3534678.3539253

[7] 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, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 165–174. https://doi.org/10.1145/3331184.3331267

[8] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 726–735. https://doi.org/10.1145/3404835.3462862

[9] Junliang Yu, Xin Xia, Tong Chen, Lizhen Cui, Nguyen Quoc Viet Hung, and Hongzhi Yin. 2024. XSimGCL: Towards Extremely Simple Graph Contrastive Learning for Recommendation. *IEEE Trans. Knowl. Data Eng.* 36, 2 (2024), 913–926. https://doi.org/10.1109/TKDE.2023.3288135

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009