Latest updates: https://dl.acm.org/doi/10.1145/3705328.3748036

SHORT-PAPER

SGCL: Unifying Self-Supervised and Supervised Learning for Graph Recommendation

WEIZHI ZHANG, University of Illinois at Chicago, Chicago, IL, United States

LIANGWEI YANG

ZIHE SONG, University of Illinois at Chicago, Chicago, IL, United States

HENRY PENG ZOU, University of Illinois at Chicago, Chicago, IL, United States

KE XU, University of Illinois at Chicago, Chicago, IL, United States

YUANJIE ZHU, University of Illinois at Chicago, Chicago, IL, United States

View all

Open Access Support provided by:

University of Illinois at Chicago

SGCL: Unifying Self-Supervised and Supervised Learning for Graph Recommendation

Weizhi Zhang
University of Illinois Chicago
Chicago, USA
wzhan42@uic.edu

Liangwei Yang*
Salesforce AI Research
Palo Alto, USA
liangwei.yang@salesforce.com

Zihe Song
University of Illinois Chicago
Chicago, USA
zsong29@uic.edu

Henry Peng Zou
University of Illinois Chicago
Chicago, USA
pzou3@uic.edu

Ke Xu
University of Illinois Chicago
Chicago, USA
kxu25@uic.edu

Yuanjie Zhu
University of Illinois Chicago
Chicago, USA
yzhu224@uic.edu

Philip S. Yu
University of Illinois Chicago
Chicago, USA
psyu@uic.edu

Abstract

Recommender systems (RecSys) are essential for online platforms, providing personalized suggestions to users within a vast sea of information. Self-supervised graph learning seeks to harness high-order collaborative filtering signals through unsupervised augmentation on the user-item bipartite graph, primarily leveraging a multi-task learning framework that includes both supervised recommendation loss and self-supervised contrastive loss. However, this separate design introduces additional graph convolution processes and creates inconsistencies in gradient directions due to disparate losses, resulting in prolonged training times and sub-optimal performance. In this study, we introduce a unified framework of Supervised Graph Contrastive Learning for recommendation (SGCL) to address these issues. SGCL uniquely combines the training of recommendation and unsupervised contrastive losses into a cohesive supervised contrastive learning loss, aligning both tasks within a single optimization direction for exceptionally fast training. Extensive experiments on three real-world datasets show that SGCL outperforms state-of-the-art methods, achieving superior accuracy and efficiency.

CCS Concepts

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

---
*Corresponding author: liangwei.yang@salesforce.com

Keywords

Graph Recommendation, Self-Supervised Learning, Contrastive Learning

1 Introduction
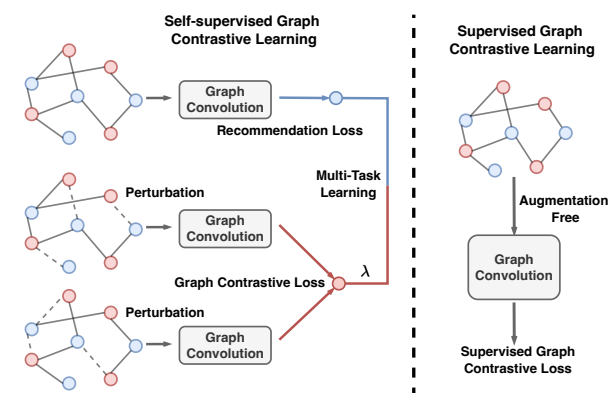
Figure 1: The left figure illustrates the pipeline of traditional SSL RecSys, including graph augmentation, three rounds of graph convolutions, and two distinct loss optimization controlled by λ. The right figure depicts our SGCL framework, consisting of a single graph encoder and a unified SGCL loss.

Recommender systems (RecSys) have emerged as a cornerstone in the architecture of modern online systems and web applications, playing a pivotal role in filtering and personalizing the information

presented to users [25, 31]. These systems are engineered to deliver tailor-made item recommendations that align with individual user preferences [35]. The scope of RecSys extends across various domains, including but not limited to, digital retailing [7, 19, 32], social networking platforms [3, 8], and video-sharing services [21]. Among the methodologies employed in RecSys, collaborative filtering (CF) stands out as a prominent approach [6, 23, 24]. CF algorithms function by extrapolating patterns from historical user-item interactions to identify and recommend items to users with similar interaction profiles. However, this approach is inherently limited by simple supervised recommendation losses and sparse user-item interactions to model and predict user preferences accurately [26, 31, 32, 34].

Graph contrastive learning has been increasingly integrated into RecSys as a solution to mitigate these challenges. This approach primarily involves data augmentation on bipartite graph structure, where high-order structural signals are extracted via graph neural networks [5, 12, 20, 33]. Concurrently, graph contrastive learning establishes a self-supervised learning (SSL) task, aimed at enhancing representation expressiveness by distinguishing the nodes in the bipartite graph. This is achieved through the creation of varied representations from different views, derived from graph augmentation techniques or embedding perturbations. Existing methodologies in this domain [1, 10, 22, 27, 30] typically employ a decoupled design approach. This involves the utilization of a recommendation-specific supervised loss function [16], designed to learn from user-item interaction signals, in conjunction with a distinct contrastive learning loss [13], which focuses on assimilating knowledge from self-supervised paradigm.

Despite their enhanced performance over traditional models that solely consider original user-item interactions [5, 20, 28], current graph contrastive learning methods in RecSys still confront **Ineffective Optimization** challenges that impede further improvements in performance and efficiency. It arises from the prevalent practice of employing a decoupled design for recommendation supervision and self-supervised contrastive signals as shown in Figure 1. This approach necessitates the redundant graph encoders to perform the graph convolutions on the whole bipartite graph, posing extreme challenges on the large-scale graph recommendation. Beyond that, the development of two distinct loss functions, while aiming to learn from different tasks, optimize the same set of parameters, leading to potential conflicts and inconsistent gradients. Such inconsistent gradients have been identified as a primary factor contributing to unstable multi-task training and performance degradation, as highlighted in recent studies [11, 17]. This not only expends considerable computational resources but also hampers further enhancement and training efficiency of the model.

In this paper, we introduce Supervised Graph Contrastive Learning for Recommendation (SGCL), a novel framework designed to address the identified obstacles. To tackle the **ineffective optimization**, SGCL incorporates a unique supervised graph contrastive learning loss. This loss function integrates the user-item interaction supervision signal directly within the self-supervised loss, thereby merging supervised and self-supervised contrastive learning tasks into a single optimization objective. Such an approach not only addresses unstable and inconsistent optimization inherent in the multi-task learning framework but also obviates the necessity of

redundant graph convolutions and extensive hyperparameter tuning of weight λ as illustrated in Figure 1. In addition, since SGCL loss computations eliminate the negative sampling process and data augmentations, the overall training is extremely time-efficient. To promote future research in supervised graph contrastive learning RecSys, we have made it available as an open-source resource https://github.com/DavidZWZ/SGCL. Our contributions are summarized as follows:

- Design of efficient framework: We have identified the **Ineffective Optimization** as a key bottleneck and designed a streamlined framework to eliminate redundant graph convolutions and complex data augmentations, marking an extremely efficient pipeline.
- Development of a novel loss: Methodologically, we have developed an innovative supervised graph contrastive learning loss, which uniquely integrates training of supervised and self-supervised tasks within a singular objective function.
- Experimentally, we conduct extensive experiments on three real-world datasets to test the effectiveness of SGCL. It achieves the highest recommendation score with remarkably low training time, demonstrating the exceptional efficiency of SGCL.

2 Preliminaries

2.1 Self-supervised Graph Contrastive Learning

Recent approaches [1, 10, 22, 27, 30] for self-supervised graph recommendation all rely on the multi-task learning framework and jointly learn from contrastive learning tasks and supervised recommendation tasks on the node representations as shown in Figure 1.

In a formal definition, the joint learning scheme in self-supervised graph recommendation is to optimize the SSLRec loss function:

$$\mathcal{L}_{\text{sslrec}} = \mathcal{L}_{\text{rec}} + \lambda \mathcal{L}_{gcl}, \tag{1}$$

where the $\mathcal{L}_{\text{sslrec}}$ consists of the traditional recommendation loss $\mathcal{L}_{\text{rec}}$, such as BPR [16], and the contrastive learning (CL) loss $\mathcal{L}_{gcl}$, with λ to weight the two losses. As the inputs are the bipartite user-item interaction graph, the contrastive learning process is conducted concurrently for both the user and item side. Based on the InfoNCE [13], the user-side CL loss $\mathcal{L}_{gcl}^{u}$ is formulated as:

$$\mathcal{L}_{gcl}^{u} = \sum_{i \in \mathcal{B}_u} -\log \frac{\exp\left(u_i'^{\top} u_i''/\tau\right)}{\sum_{k \in \mathcal{B}_u} \exp\left(u_i'^{\top} u_k''/\tau\right)}, \tag{2}$$

where i, k are the nodes (users) in the user batch $\mathcal{B}_u$ and u_i', u_i'' are two views of node representation after the data augmentations. Note that the τ is a hyperparameter that controls the temperature of the InfoNCE loss and applies the penalties on hard negative samples. Then, the item-side CL loss $\mathcal{L}_{gcl}^{v}$ can be calculated in the same format. Finally, the CL loss will be the summation of both user and item sides $\mathcal{L}_{gcl} = \mathcal{L}_{gcl}^{u} + \mathcal{L}_{gcl}^{v}$. In general, the graph contrastive learning loss $\mathcal{L}_{gcl}$ encourages the similarity between two variant embeddings u_i' and u_i'' from the same node while maximizing the inconsistency among all the other nodes in batch.

3 Proposed solutions

In this section, we introduce the proposed Supervised Graph Contrastive Learning (SGCL) in detail. Firstly, we elaborate on the novel

Supervised Graph Contrastive Loss that integrates the training of both supervised recommendation and contrastive learning losses. Then, a detailed time complexity investigation is conducted to demonstrate the efficiency of the proposed SGCL.

3.1 Supervised Graph Contrastive Learning

In the context of graph-based recommendation tasks, suppose we observe a pair of interacted users and items with corresponding initial input ID embeddings u_i^0 and v_j^0, and we adopt the light graph convolution as in [5], by consistently aggregating the connected neighbors' representations:

$$u_i^{k+1} = \sum_{j \in N_i} \frac{1}{\sqrt{|N_i|}\sqrt{|N_j|}} v_j^k,$$
$$v_j^{k+1} = \sum_{i \in N_j} \frac{1}{\sqrt{|N_i|}\sqrt{|N_j|}} u_i^k, \qquad (3)$$

where u_i^k and v_j^k are embeddings of user u_i and item v_j at k-th layer, respectively. The normalization employs the average degree $\frac{1}{\sqrt{|N_i|}\sqrt{|N_j|}}$ to temper the magnitude of popular nodes after graph convolution in each layer. Afterward, the collaborative filtering final embedding is to synthesize the layer-wise representations:

$$u_i = \sum_{k=0}^{K} \alpha_k u_i^k; \quad v_j = \sum_{k=0}^{K} \alpha_k v_j^k, \qquad (4)$$

where α_k is the representation weight of the k-th layer and K is total numbers of the layers. Note that we assume $\alpha_k = \frac{1}{K+1}$ unless mentioned in the following sections.

For the self-supervised graph recommendation, the contrastive loss $\mathcal{L}_{gcl}$ in Equation 2 is incapable of explicitly learning supervised signals from existing interactions within graph data. Therefore, recommendation loss $\mathcal{L}_{rec}$ is required for label utilization, whilst an extra weight λ should be dedicatedly tuned to balance and mitigate the gradient inconsistency. Towards uniting the power of the supervised signal and self-supervised learning strategy in the graph recommendations, we devised a new type of supervised graph contrastive learning (SGCL) loss as follows:

$$\mathcal{L}_{sgcl} = \sum_{(i,j) \in \mathcal{B}} -\log \frac{\exp\left(u_i^\top v_j / \tau\right)}{\sum_{(i',j') \in \mathcal{B}} \left(\exp(u_i^\top u_{i'} / \tau) + \exp(v_j^\top v_{j'} / \tau)\right)}, \qquad (5)$$

where (i, j) are paired user-item interaction in batch $\mathcal{B}$ and (i', j') are the rest user-item representations in batch. It is noted that this version of the loss function does not involve any type of data augmentation, and the user-item interactions are considered positive pairs. Therefore, both complex graph augmentation and time-consuming negative sampling processes are avoided, making the SGCL training process extremely efficient. More importantly, we alleviate the need to use multi-task learning for model optimization and use only one supervised graph contrastive loss during training.

3.2 Time Complexity

Here, we conduct the time complexity analysis on SGCL and compare it with predominant baseline methods LightGCN [5] and SGL [22] in graph recommendation. We first denote $|\mathcal{E}|$ as the number of edges. Then, let K be the number of layers in graph convolution, and d be the embedding size. p represents the probability of retaining the edges in SGL and N_v indicates the average number of item candidates in negative sampling.

Then, we can derive that: 1) For the normalization of the adjacency matrix, both LightGCN and SGCL do not contain the graph structure manipulation while SGL drops the edges in the graph data to create two versions of the modified adjacency matrix, and thus the time complexity is approximately three times of our methods. 2) In the graph convolution, as SGL changed the graph structure twice in contrastive learning, it requests an extra $4p|\mathcal{E}|Kd$ for graph convolution. Whereas no augmentations are implemented before graph convolution in LightGCN and SGCL, resulting in more efficient $2|\mathcal{E}|Kd$ time consumption. 3) While SGLC obviates the necessity of BPR loss computation, LightGCN and SGL still need $2BN_v$ for negative sampling and $2Bd$ for calculating the 2B pairs of interactions, respectively. 4) Regarding the graph contrastive loss, SGL separates the user and item loss calculation into $\mathcal{L}_{gcl}^u$ and $\mathcal{L}_{gcl}^v$, and thus the total time cost is $O(2Bd + 2B^2d)$. Since SGCL is based on a joint embedding learning process of the users and items, the time complexity in the batch is $O(Bd + 2B^2d)$. It is noted that, as all experiments are conducted on GPU parallel computations, therefore the time spent in loss computation differs slightly.

Table 1: Time complexity comparison of LightGCN, SGL, and SGCL in different steps of graph recommendation.

Steps	LightGCN	SGL	SGCL
Adj. Matrix	$O(2|\mathcal{E}|)$	$O((2+4p)|\mathcal{E}|)$	$O(2|\mathcal{E}|)$
Graph Conv.	$O(2|\mathcal{E}|Kd)$	$O((2+4p)|\mathcal{E}|Kd)$	$O(2|\mathcal{E}|Kd)$
Sampling	$O(BN_v)$	$O(BN_v)$	-
BPR Loss	$O(2Bd)$	$O(2Bd)$	-
GCL Loss	–	$O(2Bd + 2B^2d)$	$O(Bd + 2B^2d)$

4 Evaluation

4.1 Experimental Setup

To evaluate the superior performance and efficiency of the proposed SGCL, we conduct the experiments on two public real-world datasets from Amazon [1]: Beauty and Toys-and-Games, varying in domains and scales. We use the 5-core filtering setting to ensure the data quality for evaluation and split all datasets into training, validation, and testing with the ratio (8:1:1). In baseline selection, we compares MF methods (BiasMF [9], NeuMF [6]), graph models (NGCF [20], LightGCN [5]), self-supervised graph approaches (NCL [10], SGL [22], LightGCL [1], SimGCL [30]). For the training of all the baselines, we carefully search their hyperparameters for different datasets to ensure a fair comparison. For all approaches, batch size B and embedding size d are set to 1,024 and 64. In SGCL, the temperature τ is set to 0.2, as it was found to be optimal.

---

[1] https://jmcauley.ucsd.edu/data/amazon/links.html

Table 2: Performance comparison on Beauty and Toys-and-Games in terms of NDCG and Recall.

Method	Beauty				Toys-and-Games			
	R@20	N@20	R@50	N@50	R@20	N@20	R@50	N@50
BiasMF	0.1152	0.0544	0.1754	0.0668	0.0981	0.0479	0.1624	0.0625
NeuMF	0.0720	0.0320	0.1317	0.0490	0.0690	0.0345	0.1094	0.0428
NGCF	0.1126	0.0517	0.1699	0.0625	0.0944	0.0429	0.1532	0.0561
LightGCN	0.1224	0.0583	0.1903	0.0718	0.1159	0.0565	0.1759	0.068
NCL	0.1264	0.0613	0.1948	0.0721	0.117	0.0566	0.1782	0.0686
SGL	0.1335	0.0634	0.2048	0.078	0.1248	0.0595	0.1869	0.072
LightGCL	0.1278	0.0596	0.1905	0.0697	0.1177	0.0549	0.1772	0.0647
SimGCL	0.1345	0.064	0.2058	0.0799	0.1162	0.0593	0.1738	0.0713
SGCL	**0.141**	**0.0677**	**0.2105**	**0.0807**	**0.1317**	**0.0626**	**0.1965**	**0.0774**

## 4.2 Overall Performance Comparison

In this comprehensive experimental study, we evaluated the performance of several state-of-the-art recommendation methods across diverse datasets: Beauty and Toys-and-Games, using Recall@20, Recall@50, NDCG@20, and NDCG@50. Predominantly, SGCL achieved the highest NDCG and Recall scores across all datasets, highlighting its superior efficacy in recommendation tasks. This outcome not only emphasizes the significance of the augmentation-free framework but also validates the effectiveness of the supervised graph contrastive loss optimization. Most of the self-supervised graph recommender systems consistently outperform the traditional ones. This suggests that the auxiliary contrastive learning task leverages extra graph structure information, boosting the performance of the predictions on unobserved user-item interactions.

4.3 Efficiency Analysis

4.3.1 *Trade-off between the Performance and the efficiency.*
Figure 2 demonstrates the superiority of our models in both efficiency and performance. Two matrix factorization models are more efficient than graph models but achieve much lower performance. LightGCN and NGCF outperform the rest of the graph contrastive learning baselines in terms of speed. Notably, our method SGCL exhibits the most competitive performance while being much faster than the other graph contrastive recommendation baselines.

4.3.2 *Training Time Comparison.* In Table 3, LightGCN is efficient per epoch but may not always be the best choice when considering total training time due to the higher number of epochs required. SGL appears to offer a better compromise between the number of epochs and time per epoch, often resulting in an approximate total time compared with LightGCN. In contrast, SGCL demonstrates extreme efficiency in all datasets, which run for the fewest epochs despite being the fastest per epoch in most cases, thereby resulting in the lowest total training time in all datasets.

4.3.3 *Performance Curve and Convergence Speed.* We compare the NDCG@20 (Figure 3) across the first 30 training epochs with the LightGCN and the SGL on three datasets. By 30 epochs, both of our model SGCL achieve high and stable Recall levels,
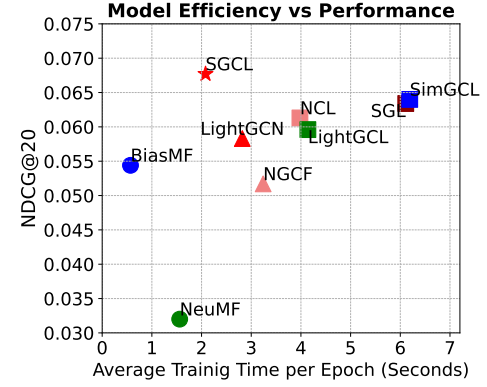
Figure 2: Trade-off between the performance and the efficiency on the Beauty dataset. The upper side indicates better performance; the left side represents more efficient training.

Table 3: Training time efficiency comparisons (we denote second as s, and minute as m as the abbreviation).

Dataset	Method	Time/Epoch	# Epochs	Total Time
Beauty	LightGCN	2.82s	153	7.19m
	SGL	6.11s	63	6.42m
	SGCL	2.08s	62	2.15m
Toys-and-Games	LightGCN	1.53s	173	4.41m
	SGL	6.24s	75	7.80m
	SGCL	1.63s	35	0.95m

whereas the LightGCN and the SGL have been far from convergence points. In consonance with the outcomes of our training time experiments in Figure 2, SGCL demonstrates low total training time and fast convergence speed. These findings underscore the preeminence of SGCL in practical and real-world applications.
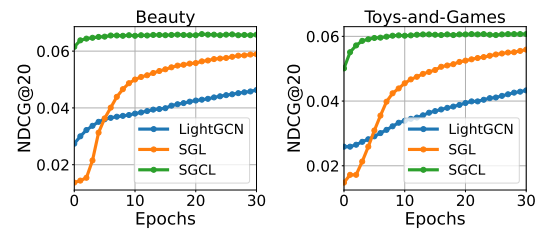
Figure 3: Performance of NDCG in the first 30 epochs.

5 Related Work

Recently, self-supervised contrastive learning has gained considerable attention across various fields due to its proficiency in handling massive amounts of unlabeled data for the data sparsity issue [2, 4, 14, 18, 29]. Motivated by this, NCL[10] incorporates potential neighbors into contrastive pairs, by considering both the graph structure and semantic space. Differently, the dropout-based methods are employed by SGL[22] for randomized graph augmentation

to obtain two contrastive views. The optimization includes node-level contrast using the InfoNCE loss, along with the BPR loss for recommendation. SimGCL[30] first investigates the necessity of complex graph augmentation and then devises noise-based augmentation techniques that perturb the representations of users and items during training. Then, LightGCL [1] has been proposed to utilize the singular value decomposition (SVD) [15] as a light tool to augment user-item graph structures. Nevertheless, all the previous works follow the multi-task learning paradigm with tedious manual tuning of task weight and require additional graph encoders as well as different augmentation procedures during training.

6 Conclusion

In this paper, we revisit the existing graph contrastive learning methodologies in the recommendation and identify ***Ineffective Optimization*** challenge. Towards this issue, SGCL innovatively integrates the training of recommendation and contrastive losses using a supervised contrastive learning loss. This unified approach effectively resolves the problem of inconsistent optimization directions and the inefficient graph convolutions, typically seen in decoupled multi-task learning frameworks. The empirical validation of SGCL, conducted through extensive experiments on real-world datasets, underscores its superior performance. Notably, SGCL outshines state-of-the-art methods, achieving faster convergence speed, and excelling in both accuracy and efficiency. This paper's findings highlight the potential of coupled design in supervised graph contrastive learning, paving the way for more efficient and effective recommender systems.

Acknowledgments

References

[1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *The Eleventh International Conference on Learning Representations*.

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

[3] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The world wide web conference*. 417–426.

[4] Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International conference on machine learning*. PMLR, 4116–4126.

[5] 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.

[6] 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.

[7] Hyunwoo Hwangbo, Yang Sok Kim, and Kyung Jin Cha. 2018. Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications* 28 (2018), 94–101.

[8] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. 135–142.

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[10] 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.

[11] Qi Liu, Zhilong Zhou, Gangwei Jiang, Tiezheng Ge, and Defu Lian. 2023. Deep Task-specific Bottom Representation Network for Multi-Task Recommendation. *arXiv preprint arXiv:2308.05996* (2023).

[12] Jing Ma, Liangwei Yang, Qiong Feng, Weizhi Zhang, and Philip S Yu. 2023. Graph-based village level poverty identification. In *Proceedings of the ACM web conference 2023*. 4115–4119.

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

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

[15] Ajit Rajwade, Anand Rangarajan, and Arunava Banerjee. 2012. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 4 (2012), 849–862.

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

[17] Jiaxi Tang, Yoel Drori, Daryl Chang, Maheswaran Sathiamoorthy, Justin Gilmer, Li Wei, Xinyang Yi, Lichan Hong, and Ed H Chi. 2023. Improving Training Stability for Multitask Ranking Models in Recommender Systems. *arXiv preprint arXiv:2302.09178* (2023).

[18] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. 2020. What makes for good views for contrastive learning? *Advances in neural information processing systems* 33 (2020), 6827–6839.

[19] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 645–653.

[20] 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.

[21] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *Proceedings of the ACM Web Conference 2023*. 790–800.

[22] 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.

[23] Ke Xu, Weizhi Zhang, Zihe Song, Yuanjie Zhu, and Philip S Yu. 2025. Graph Neural Controlled Differential Equations For Collaborative Filtering. In *Companion Proceedings of the ACM on Web Conference 2025*. 1446–1449.

[24] Ke Xu, Yuanjie Zhu, Weizhi Zhang, and S Yu Philip. 2023. Graph Neural Ordinary Differential Equations-based method for Collaborative Filtering. In *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1445–1450.

[25] Wooseong Yang, Chen Wang, Zihe Song, Weizhi Zhang, and Philip S Yu. 2024. Item Cluster-aware Prompt Learning for Session-based Recommendation. *arXiv preprint arXiv:2410.04756* (2024).

[26] Wooseong Yang, Weizhi Zhang, Yuqing Liu, Yuwei Han, Yu Wang, Junhyun Lee, and Philip S Yu. 2025. Cold-Start Recommendation with Knowledge-Guided Retrieval-Augmented Generation. *arXiv preprint arXiv:2505.20773* (2025).

[27] Yonghui Yang, Zhengwei Wu, Le Wu, Kun Zhang, Richang Hong, Zhiqiang Zhang, Jun Zhou, and Meng Wang. 2023. Generative-contrastive graph learning for recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1117–1126.

[28] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[29] 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.

[30] 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.

[31] Weizhi Zhang, Yuanchen Bei, Liangwei Yang, Henry Peng Zou, Peilin Zhou, Aiwei Liu, Yinghui Li, Hao Chen, Jianling Wang, Yu Wang, et al. 2025. Cold-start recommendation towards the era of large language models (llms): A comprehensive survey and roadmap. *arXiv preprint arXiv:2501.01945* (2025).

[32] Weizhi Zhang, Liangwei Yang, Yuwei Cao, Ke Xu, Yuanjie Zhu, and S Yu Philip. 2023. Dual-Teacher Knowledge Distillation for Strict Cold-Start Recommendation. In *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 483–492.

[33] Weizhi Zhang, Liangwei Yang, Zihe Song, Henry Peng Zou, Ke Xu, Liancheng Fang, and Philip S Yu. 2024. Do we really need graph convolution during training? light post-training graph-ode for efficient recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 3248–3258.

[34] Weizhi Zhang, Liangwei Yang, Wooseong Yang, Henry Peng Zou, Yuqing Liu, Ke Xu, Sourav Medya, and Philip S Yu. 2025. Llminit: A free lunch from large language models for selective initialization of recommendation. *arXiv preprint arXiv:2503.01814* (2025).

[35] Weizhi Zhang, Xinyang Zhang, Chenwei Zhang, Liangwei Yang, Jingbo Shang, Zhepei Wei, Henry Peng Zou, Zijie Huang, Zhengyang Wang, Yifan Gao, et al. 2025. Personaagent: When large language model agents meet personalization at test time. *arXiv preprint arXiv:2506.06254* (2025).