Graph-based Alignment and Uniformity for Recommendation

Liangwei Yang
University of Illinois Chicago
Chicago, USA
lyang84@uic.edu

Zhiwei Liu
Salesforce AI Research
Palo Alto, USA
zhiweiliu@salesforce.com

Chen Wang
University of Illinois Chicago
Chicago, USA
cwang266@uic.edu

Mingdai Yang
Xiaolong Liu
myang72@uic.edu
xliu262@uic.edu
University of Illinois at Chicago
Chicago, USA

Jing Ma
University of Electronic Science and Technology of China
Chengdu, China
jingma@uestc.edu.cn

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

ABSTRACT

Collaborative filtering-based recommender systems (RecSys) rely on learning representations for users and items to predict preferences accurately. Representation learning on the hypersphere is a promising approach due to its desirable properties, such as alignment and uniformity. However, the sparsity issue arises when it encounters RecSys. To address this issue, we propose a novel approach, graph-based alignment and uniformity (GraphAU), that explicitly considers high-order connectivities in the user-item bipartite graph. GraphAU aligns the user/item embedding to the dense vector representations of high-order neighbors using a neighborhood aggregator, eliminating the need to compute the burdensome alignment to high-order neighborhoods individually. To address the discrepancy in alignment losses, GraphAU includes a layer-wise alignment pooling module to integrate alignment losses layer-wise. Experiments on four datasets show that GraphAU significantly alleviates the sparsity issue and achieves state-of-the-art performance. We open-source GraphAU at https://github.com/YangLiangwei/GraphAU.

CCS CONCEPTS

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

KEYWORDS

Graph Neural Network, Recommendation System, Alignment

ACM Reference Format:
Liangwei Yang, Zhiwei Liu, Chen Wang, Mingdai Yang, Xiaolong Liu, Jing Ma, and Philip S. Yu. 2023. Graph-based Alignment and Uniformity for Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3583780.3615185

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615185

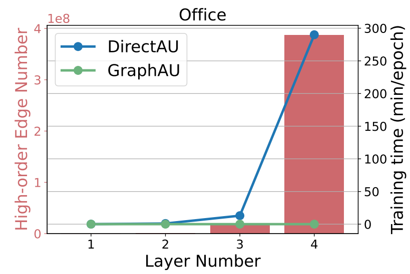

Figure 1: The scalability problem of DirectAU, and the efficiency of GraphAU. GraphAU nearly keeps constant Training time (line plot) with High-order Edge Number (Bar Plot).

1 INTRODUCTION

In regard to the overwhelming online information [17], recommender systems (RecSys) assist users effortlessly discovering items that align with their interests [6, 15, 30, 32]. The core of most existing RecSys is learning representations for users and items. By embedding past user-item interactions into dense vectors [33], RecSys create sophisticated representations of users and items that capture the nuances of their preferences and characteristics. The effectiveness of learning user/item representations is heavily influenced by the choice of loss functions [1, 11, 20, 22, 23, 31]. Recent studies explain that the desirable performance of contrastive loss results from the alignment of positive pairs and uniformity of data [4, 19, 27]. This observation motivates the development of an innovative technique, Direct Alignment and Uniformity (DirectAU) [25] in the field of RecSys. DirectAU employs alignment loss to improve the normalized element-wise similarity between a user's representation and those of the items he/she has interacted with. It also adopts uniformity loss to ensure that the user and item representations are evenly distributed and inherently distinguishable from each other.

Despite the effectiveness, we contend that directly utilizing alignment between user-item pairs, DirectAU neglects the critical data sparsity [7] issue in RecSys. The alignment signals between users and his/her direct interacted items are rather sparse, which hinders the effectiveness of alignment objective. Inspired by existing works [2, 9, 14, 21, 26, 28, 29, 34, 35] that leverage high-order connections in user-item bipartite graph, we propose to devise a novel GraphAU to enhance current alignment loss. The direct alignment

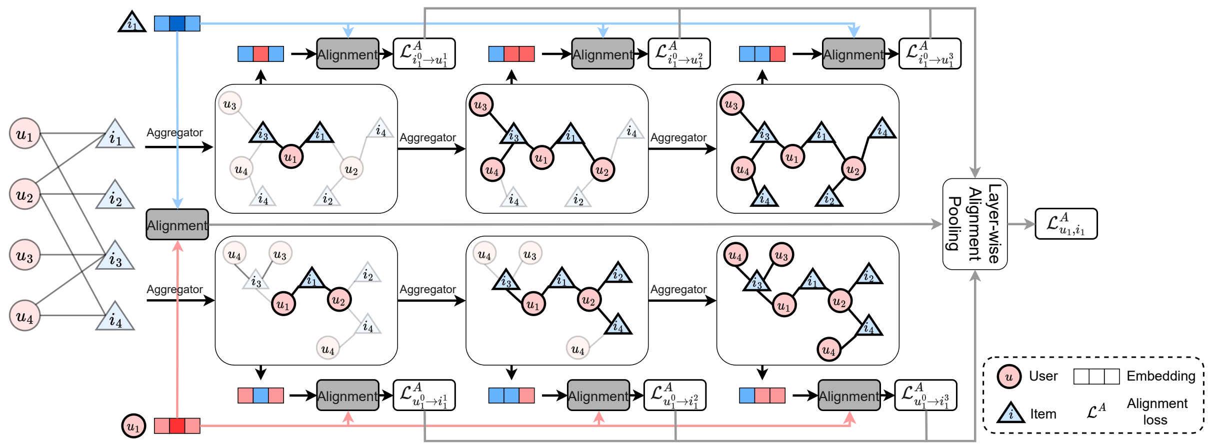

Figure 2: An example of computing alignment loss between u_1 and i_1 considering neighborhoods within 3 hops. GraphAU applies 3 Aggregators on the user-item bipartite graph to obtain a dense neighborhoods representation of u_1/i_1 within hops from 1 to 3. The output of each Aggregator is explicitly aligned with u_1 or i_1 correspondingly before the pooling module.

of high-order connectivities encounters two primary challenges. Firstly, the scalability of high-order connections poses a significant hurdle. As shown in Figure 1, the number of high-order edges increases exponentially with more hops, and the training time of DirectAU also increases exponentially. Consequently, direct alignment of all high-order edges can be impractical and time-consuming. Secondly, there is a discrepancy in different orders. Alignment losses from different hop neighborhoods exhibit neighborhood similarity and influence scope discrepancies. Low-order neighborhoods in the user-item bipartite graph are typically more pertinent to the center node. Besides, the influence of alignment loss from low-order neighbors is only propagated to a small portion of nodes compared with high-order neighbors. It is necessary to consider the discrepancies of alignment losses in different orders.

This paper proposes a novel approach, called graph-based alignment and uniformity (GraphAU), to address the sparsity issue by explicitly considering high-order connectivities in the user-item bipartite graph. To overcome the scalability issue, GraphAU aligns the user/item embedding to the dense vector representations of high-order neighbors instead of directly aligning to high-order neighborhoods individually. To achieve this, several layers of aggregators are used to obtain the dense representation of neighborhoods within different hops. Then, the user/item embedding is directly aligned to the dense representation of high-order neighborhoods of the connected item/user. This approach eliminates the need for individually computing the burdensome alignment to high-order neighborhoods and resolves the scalability issue through neighborhood aggregation. As shown in Figure 1, the training time per epoch of GraphAU nearly keeps constant with the considered exponential increased high-order edges. Compared with DirectAU, GraphAU greatly reduces the training time and enables the alignment toward high-order neighborhoods. To address the discrepancy in alignment losses, GraphAU includes a layer-wise alignment pooling module to integrate alignment losses layer-wise. A modification factor α is introduced to adjust the weight of alignment loss from different layers. GraphAU significantly alleviates the sparsity issue and achieves state-of-the-art performance on four datasets with varying scales. The contributions of our paper are summarized as follows:

- We propose a novel approach, named GraphAU, that explicitly considers high-order connectivities in the user-item bipartite graph to address the sparsity issue.
- To address the discrepancy in alignment losses, we propose a simple and effective layer-wise alignment pooling module to integrate alignment losses layer-wise.
- We conduct extensive experiments on 4 real-world datasets with varying scales and demonstrate the effectiveness of GraphAU.

2 PRELIMINARIES

Problem Statement. In the context of a recommendation task, the objective is to generate a list of top k items that a given user u is likely to be interested in. This is based on the historical interactions between users and items, which are represented by an interaction matrix $\mathbf{R}$ of size $|\mathcal{U}| \times |\mathcal{I}|$. The set of users is denoted by $\mathcal{U} = \{u_1, u_2, ..., u_{|\mathcal{U}|}\}$ and the set of items by $\mathcal{I} = \{i_1, i_2, ..., i_{|\mathcal{I}|}\}$. In this paper, we only consider the implicit feedback, and the interaction matrix $\mathbf{R}$ is binary, where $R_{u,i} = 1$ if user u has interacted with item i, and $R_{u,i} = 0$ otherwise. To model the historical interactions, a user-item bipartite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed, where $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$ and there is an edge $(u, i) \in \mathcal{E}$ between u and i if $R_{u,i} = 1$. The objective of the recommender system is to learn from this graph $\mathcal{G}$ and generate a ranked list of potential items for each user u.

Alignment and Uniformity in RecSys. The two properties [5, 27] are introduced into RecSys by DirectAU [25] that considers the user-item interactions as positive pairs. They are optimized by two different losses. Alignment loss is calculated by:

$$\mathcal{L}^A = \frac{1}{|\mathcal{E}|} \sum_{(u,i)\in\mathcal{E}} \mathcal{L}^A_{u,i} = \frac{1}{|\mathcal{E}|} \sum_{(u,i)\in\mathcal{E}} ||\mathbf{e}_u - \mathbf{e}_i||^2, \tag{1}$$

where $\mathcal{L}^A_{u,i}$ is the alignment loss from u to i, and $\mathbf{e}_u/\mathbf{e}_i$ is user/item representation. Alignment aims to strengthen the normalized element-wise similarity between the user's and his/her interacted items' representation. Uniformity loss is calculated as $\mathcal{L}^U = \frac{1}{2}(\mathcal{L}^U_{\mathcal{U}} + \mathcal{L}^U_{\mathcal{I}})$, where $\mathcal{L}_{\mathcal{U}}^{U}$ is the uniformity loss for user set $\mathcal{U}$, denoted as:

$$\mathcal{L}_{\mathcal{U}}^{U} = \log \frac{1}{|\mathcal{U}|^2} \sum_{u \in \mathcal{U}} \sum_{u^* \in \mathcal{U}} e^{-2||e_u - e_u^*||}. \tag{2}$$

Item set uniformity $\mathcal{L}_{\mathcal{I}}^{U}$ is calculated analogously. Uniformity targets distributing user/item representation uniformly and distinguishable. It is utilized as a regularization to avoid a trivial solution (*i.e.* all same) of users/item embedding through alignment loss.

3 PROPOSED MODEL

The framework of GraphAU is shown in Figure 2. Two designed modules are described as follows.

3.1 Multi-hop Neighborhood Alignment

The proposed Multi-hop Neighborhood Alignment Module addresses the scalability issue of aligning high-order neighborhoods. This is achieved through an aggregator to obtain dense vector representations of neighborhoods within multiple hops. By doing so, the scalability issue is addressed. The module directly computes the alignment loss towards the high-order dense neighborhood representations, eliminating the need to align high-order neighborhoods, which can be burdensome individually.

Similar to the approach used in learning representations of words and graphs [13, 24], embedding techniques have been widely adopted in recommender systems [9, 12, 20]. This involves the use of an embedding layer, which serves as a look-up table to map user and item IDs to dense vectors, denoted as $\mathbf{E}^{(0)} = \left(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_{|\mathcal{U}|+|\mathcal{I}|}^{(0)}\right)$, where $\mathbf{e}^{(0)} \in \mathbb{R}^d$ is a d-dimensional dense vector corresponding to a specific user or item. The embedding is then fed into an aggregator for information aggregation. The resulting output from the embedding layer is commonly referred to as the 0-th layer output, denoted as $\mathbf{e}_i^{(0)}$. We then compute the dense vector representation of neighborhoods via an aggregator as:

$$\mathbf{e}_u^{(l+1)} = \text{Aggregator}^{(l+1)}(\{\mathbf{e}_i^{(l)} \mid i \in \mathcal{N}_u\}), \tag{3}$$

where $\mathbf{e}_u^{(l)}$ indicates node u's embedding on the l-th layer, $\mathcal{N}_u$ is the neighbor set of node u, $\text{Aggregator}^{(l)}(\cdot)$ aggregates neighbors' embeddings into a single vector for layer l. The aggregator for items is computed similarly. The aggregator can be any pooling function over the neighborhood's representation to a dense vector as long as it is non-parametric. Each aggregation would generate one embedding vector for each user/item node. Embedding generated from different layers is the dense representation of neighbors within different hops. After L-th layer convolution, we can have the multi-hop neighborhood alignment loss as Equation 1 from different layers, as illustrated in Figure 2:

$$\begin{aligned} &(\mathcal{L}^A_{u^{(0)},i^{(1)}}, \mathcal{L}^A_{u^{(0)},i^{(2)}}, ..., \mathcal{L}^A_{u^{(0)},i^{(L)}}), \\ &(\mathcal{L}^A_{i^{(0)},u^{(1)}}, \mathcal{L}^A_{i^{(0)},u^{(2)}}, ..., \mathcal{L}^A_{i^{(0)},u^{(L)}}), \end{aligned} \tag{4}$$

where $\mathcal{L}^A_{u^{(0)},i^{(L)}} = ||\mathbf{e}_u^{(0)} - \mathbf{e}_i^{(L)}||^2$ is the alignment loss from u's embedding $\mathbf{e}_u^{(0)}$ to the i's representation after L layers aggregator $\mathbf{e}_i^{(L)}$, which is the dense representation of neighbors within L hops. GraphAU aligns the embedding layer directly to high-order neighborhood representations, facilitating the embedding layer's awareness of the high-order information of the counterpart.

3.2 Layer-wise Alignment Pooling

The alignment losses obtained from different layers vary regarding their relevance to the center node and the scope of their influence. To address this, a layer-wise alignment pooling module is proposed to integrate the discrepant losses. Firstly, the user and item alignment losses from the same layer are combined through averaging: $\mathcal{L}_{u,i}^{A,(L)} = \frac{1}{2}(\mathcal{L}^A_{u^{(0)},i^{(L)}} + \mathcal{L}^A_{i^{(0)},u^{(L)}})$. We then use a modification factor α to adjust the alignment weight for each layer and combine them with a weighted sum:

$$\mathcal{L}_{u,i}^{A} = \mathcal{L}_{u,i}^{A,(0)} + \alpha\mathcal{L}_{u,i}^{A,(1)} + \cdots + \alpha^L\mathcal{L}_{u,i}^{A,(L)}, \tag{5}$$

where $\mathcal{L}_{u,i}^{A,(0)} = ||\mathbf{e}_u^{(0)} - \mathbf{e}_i^{(0)}||^2$ is the direct alignment loss. GraphAU incorporates a weight factor, denoted by α, to balance the importance of relevant and high-order alignment losses. Specifically, when $\alpha < 1$, the method emphasizes the relevant alignment loss and gradually assigns less weight to high-order alignment loss. When $\alpha = 1$, no discrepancy is made for different layers. Finally, when $\alpha > 1$, the method prioritizes the influence scope and assigns more weight to the high-order alignment loss.

A uniformity loss is included to prevent all the embedding from aligning identically and enable easier distinction between users and items. The final loss function is as follows:

$$\mathcal{L} = \frac{1}{|\mathcal{E}|} \sum_{(u,i) \in \mathcal{E}} \mathcal{L}_{u,i}^{A} + \frac{\gamma}{2} \cdot (\mathcal{L}_{\mathcal{U}}^{U} + \mathcal{L}_{\mathcal{I}}^{U}), \tag{6}$$

where γ is a trade-off hyper-parameter for alignment and uniformity. After optimization, the rating score from u to i is directly computed as their embedding dot product as $R_{u,i} = \mathbf{e}_u^{(0)\top} \mathbf{e}_i^{(0)}$.

4 EXPERIMENTS

4.1 Experimental Setup

Datasets: We conduct experiments on 4 real-world datasets with varying scales, including Amazon Office, Toys, and Beauty datasets [8, 18] and one check-in dataset Gowalla [3]. We randomly split the dataset into training set (60%), validation set (20%), and test set (20%). We select Recall@20 (R@20), Hit Ratio@20 (HR@20) and nDCG@20 (N@20) as metrics. **Baselines**: To justify the effectiveness of GraphAU, we compare it with 5 baseline methods. MF-BPR [20], NGCF [28], LightGCN [9], UltraGCN [16] and DirectAU [25]. To make a fair comparison, we fix the embedding size as 32, and tune all the methods based on the same grid search. The learning rate is searched in $\{0.1, 0.05, 0.01, 0.005\}$. We use Adam optimizer [10] and search weight decay in $\{0.0, 1e^{-2}, 1e^{-4}, 1e^{-6}, 1e^{-8}\}$. For NGCF, LightGCN, and GraphAU, layer number is tuned from 1 to 4. For DirectAU and GraphAU, γ is tuned from 0.0 to 1.0 with a step of 0.1. For GraphAU, we use Light Graph Convolution [9] as the aggregator and tune α from 0.0 to 2.0 with a step of 0.1. We apply early stop with 10 epochs to avoid over-fitting.

4.2 Performance Evaluation

Experiment results are shown in Table 1. We can have the following observations. 1) GraphAU achieves the best performance on

Table 1: Overall comparison, the best and second-best results are in bold and underlined, respectively

Method	Office			Toys			Beauty			Gowalla		
	R@20	HR@20	N@20	R@20	HR@20	N@20	R@20	HR@20	N@20	R@20	HR@20	N@20
MF-BPR	0.0818	0.1762	0.0441	0.0755	0.1193	0.0393	0.0580	0.0936	0.0303	0.0824	0.3213	0.0627
NGCF	0.0753	0.1689	0.0403	0.0668	0.1048	0.0346	0.0843	0.1327	0.0453	0.0856	0.3288	0.0648
LightGCN	0.0711	0.1554	0.0380	0.0937	0.1483	0.0490	0.1035	0.1628	0.0546	0.0720	0.2901	0.0552
UltraGCN	0.0796	0.1639	0.0439	0.0928	0.1481	0.0493	0.1043	0.1619	0.0560	0.0887	0.3411	0.0668
DirectAU	0.0839	0.1682	0.0474	0.0985	0.1559	0.0531	0.1074	0.1677	0.0583	0.1134	0.4051	0.0847
GraphAU	**0.0979**	**0.2003**	**0.0539**	**0.1041**	**0.1637**	**0.0551**	**0.1124**	**0.1752**	**0.0599**	**0.1174**	**0.4136**	**0.0855**
Improvement	16.64%	13.68%	13.71%	5.71%	5.04%	3.81%	4.63%	4.47%	2.78%	3.52%	2.11%	1.01%

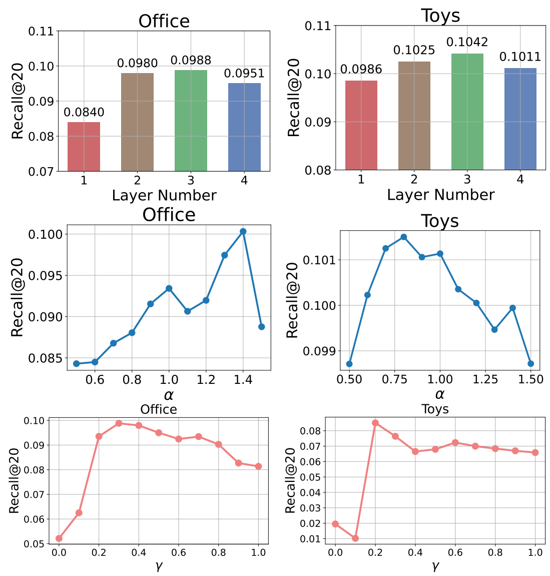

Figure 3: Parameter sensitivity of layer number, α and γ

all the datasets. On the Office dataset, GraphAU surpasses DirectAU for 16.64% in R@20. It justifies the advantages of considering high-order alignment in RecSys. 2) DirectAU performs better than other graph-based baselines. It shows a direct utilization of alignment and uniformity on user-item pairs can learn an informative user/item embedding, which justifies the advantages of alignment and uniformity-based user/item representation learning in RecSys. 3) The performance of graph-based baselines varies with different datasets. For example, NGCF overpasses LightGCN on Office and Gowalla datasets while failing on Toys and Beauty dataset. It shows pure graph-based methods are sensitive to datasets.

4.3 Model Analysis

We further analyze the influential hyper-parameters layer number L, modification factor α, and alignment/uniformity trade-off γ of GraphAU. Experiment results are shown in Figure 3. We can have the following observations. 1) L: With the increase of layer number, the performance of GraphAU first increases to its peak and then decreases on both datasets. The initial increase shows GraphAU alleviates the sparsity issue by aligning high-order neighborhoods' representation, while the following decrease indicates excessive high-order alignment leads to the performance drop. 2) α: When

Table 2: High-order uniformity experiment on Office.

Method	R@20	HR@20	N@20
GraphAU	**0.0979**	**0.2003**	**0.0539**
+ 1-st order uniformity	0.0849	0.1704	0.0482
+ 2-nd order uniformity	0.0857	0.1732	0.0473
+ 3-rd order uniformity	0.0799	0.1600	0.0440

we conduct experiments on α, we fix the number of layers as 4 to observe its influence better. Experiments show α impacts the performance of GraphAU greatly. An interesting finding is that the peak point of α can be smaller (as in Toys) or larger (as in Office) than 1 based on different datasets. It shows for different datasets, GraphAU adapts on different aspects. When $\alpha < 1$, GraphAU assigns decaying weights for high-order alignment to focus more on local relevant neighborhoods. When $\alpha > 1$, GraphAU increases the weights for high-order alignment, which shows it focuses more on the influence scope of alignment loss. 3) γ: It is the most influential hyper-parameter on all datasets, which requires careful tuning for GraphAU. Similar to high-order alignment in GraphAU, we also investigate the impact of high-order uniformity by adding uniformity loss on the aggregated user/item high-order embedding. Results are shown in Table 2. We can see that adding high-order uniformity loss deteriorates the performance of GraphAU. It is because the graph explicitly represents only the alignment information between users and items. Enforcing excessive regularization of high-order uniformity is not a favorable option for GraphAU.

5 CONCLUSION

In this paper, we identify, study, and cope with the sparsity issue when alignment/uniformity encounters RecSys. We propose a solution called GraphAU, which focuses on explicitly aligning user/item embeddings while taking into account high-order connectivities in the user-item bipartite graph. To address scalability, GraphAU aligns towards the dense vector neighborhood representation obtained by an aggregator, rather than aligning with high-order neighborhoods individually. We also introduce a modification factor that effectively integrates discrepant alignment losses from different layers. Our experiments demonstrate the benefits of aligning high-order neighborhoods and the effectiveness of GraphAU.

ACKNOWLEDGMENTS

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

[1] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*. 129–136.

[2] 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.

[3] 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.

[4] Benoit Dufumier, Pietro Gori, Julie Victor, Antoine Grigis, and Edouard Duchesnay. 2021. Conditional Alignment and Uniformity for Contrastive Learning with Continuous Proxy Labels. *arXiv preprint arXiv:2111.05643* (2021).

[5] Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821* (2021).

[6] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical user profiling for e-commerce recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 223–231.

[7] Guibing Guo, Huihuai Qiu, Zhenhua Tan, Yuan Liu, Jing Ma, and Xingwei Wang. 2017. Resolving data sparsity by multi-type auxiliary implicit feedback for recommender systems. *Knowledge-Based Systems* 138 (2017), 202–207.

[8] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[9] 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.

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

[11] Zelong Li, Jianchao Ji, Yingqiang Ge, and Yongfeng Zhang. 2022. AutoLossGen: Automatic Loss Function Generation for Recommender Systems. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1304–1315.

[12] Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip S Yu. 2021. Dense hierarchical retrieval for open-domain question answering. *arXiv preprint arXiv:2110.15439* (2021).

[13] Ye Liu, Lifang He, Bokai Cao, Philip Yu, Ann Ragin, and Alex Leow. 2018. Multi-view multi-graph embedding for brain network clustering analysis. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[14] Ye Liu, Yao Wan, Lifang He, Hao Peng, and S Yu Philip. 2021. Kg-bart: Knowledge graph-augmented bart for generative commonsense reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 6418–6425.

[15] Linyuan Lü, Matúš Medo, Chi Ho Yeung, Yi-Cheng Zhang, Zi-Ke Zhang, and Tao Zhou. 2012. Recommender systems. *Physics reports* 519, 1 (2012), 1–49.

[16] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[17] Viktor Mayer-Schönberger and Kenneth Cukier. 2013. *Big data: A revolution that will transform how we live, work, and think.* Houghton Mifflin Harcourt.

[18] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[19] Shi Pu, Kaili Zhao, and Mao Zheng. 2022. Alignment-Uniformity aware Representation Learning for Zero-shot Video Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19968–19977.

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

[21] Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. 2021. How powerful is graph convolution for recommendation?. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 1619–1629.

[22] Hao Tang, Guoshuai Zhao, Yujiao He, Yuxia Wu, and Xueming Qian. 2023. Ranking-based contrastive loss for recommendation systems. *Knowledge-Based Systems* 261 (2023), 110180.

[23] Hao Tang, Guoshuai Zhao, Yuxia Wu, and Xueming Qian. 2021. Multisample-based contrastive loss for top-k recommendation. *IEEE Transactions on Multimedia* (2021).

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[25] 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 *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1816–1825.

[26] Shoujin Wang, Liang Hu, Yan Wang, Xiangnan He, Quan Z Sheng, Mehmet A Orgun, Longbing Cao, Francesco Ricci, and Philip S Yu. 2021. Graph learning based recommender systems: A review. *arXiv preprint arXiv:2105.06339* (2021).

[27] Tongzhou Wang and Phillip Isola. 2020. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*. PMLR, 9929–9939.

[28] 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.

[29] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled graph collaborative filtering. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*. 1001–1010.

[30] Chuhan Wu, Fangzhao Wu, Tao Qi, Qi Liu, Xuan Tian, Jie Li, Wei He, Yongfeng Huang, and Xing Xie. 2022. Feedrec: News feed recommendation with various user feedbacks. In *Proceedings of the ACM Web Conference 2022*. 2088–2097.

[31] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S Yu. 2021. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th international ACM SIGIR conference on Research and development in information retrieval*. 2141–2145.

[32] Liangwei Yang, Zhiwei Liu, Yu Wang, Chen Wang, Ziwei Fan, and Philip S Yu. 2022. Large-scale personalized video game recommendation via social-aware contextualized graph neural network. In *Proceedings of the ACM Web Conference 2022*. 3376–3386.

[33] Liangwei Yang, Shengjie Wang, Yunzhe Tao, Jiankai Sun, Xiaolong Liu, Philip S Yu, and Taiqing Wang. 2022. DGRec: Graph Neural Network for Recommendation with Diversified Embedding Generation. *arXiv preprint arXiv:2211.10486* (2022).

[34] Mingdai Yang, Zhiwei Liu, Liangwei Yang, Xiaolong Liu, Chen Wang, Hao Peng, and Philip S Yu. 2023. Ranking-based group identification via factorized attention on social tripartite graph. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 769–777.

[35] Yiding Zhang, Chaozhuo Li, Xing Xie, Xiao Wang, Chuan Shi, Yuming Liu, Hao Sun, Liangjie Zhang, Weiwei Deng, and Qi Zhang. 2022. Geometric Disentangled Collaborative Filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 80–90.