Dynamic Graph Construction for Improving Diversity of Recommendation

Rui Ye*
Yuqing Hou*
yereui03@meituan.com
houyuqing@meituan.com
Meituan
Beijing, China

Te Lei
Meituan
Beijing, China
leite20@mails.tsinghua.edu.cn

Yunxing Zhang
Meituan
Beijing, China
zhangyunxing@meituan.com

Jiale Guo
Meituan
Beijing, China
guojiale04@meituan.com

Huaiwen Wu
Meituan
Beijing, China
wuhuaiwen@meituan.com

Qing Zhang
Meituan
Beijing, China
zhangqing31@meituan.com

Hengliang Luo
Meituan
Beijing, China
luohengliang@meituan.com

ABSTRACT

The diversity of recommendation has attracted a lot of attention in recommender systems due to its ability to improve user experience. Most of the diversified recommendation tasks usually exploit user-item interaction records for mining user explicit preferences, while rarely explore the user-item non-interaction records. For diversified recommendations, however, the neglected non-interaction records are especially important for capturing users' potential interests to improve the diversity of recommendation. Moreover, the majority of diversified recommendation methods run in two stages: first optimizing the users and items embeddings by relevance, then generating the diversified items list by post-processing methods. These methods are not end-to-end thus can hardly reach global optimum. To solve above limitations, we propose an end-to-end Dynamic Diversified Graph framework (DDGraph) which constructs the user-item graph dynamically based on the users and items embeddings. Technically, we initialize a user-item interaction graph and dynamically update the graph by selecting a set of diverse items for each user and building links between the items and user. The selection of diverse items can be achieved by different candidate selection operators. Specifically, we design a Quantile Progressive Candidate Selection (QPCS) operator based on the latent space division. To the best of our knowledge, our method is the first to diversify recommendation results by dynamic end-to-end graph construction and the QPCS has a higher computational efficiency than other operators. Extensive experiments on the benchmark dataset illustrate the effectiveness and superiority of the DDGraph framework.

*Both authors contributed equally to this research.

RecSys '21, September 27-October 1, 2021, Amsterdam, Netherlands

ACM ISBN 978-1-4503-8458-2/21/09.
https://doi.org/10.1145/3460231.3478845

KEYWORDS

Recommender System, Diversified Recommendation, Graph Neural Networks, Dynamic Graph Construction, Quantile Progressive Candidate Selection

ACM Reference Format:
Rui Ye, Yuqing Hou, Te Lei, Yunxing Zhang, Jiale Guo, Huaiwen Wu, Qing Zhang, and Hengliang Luo . 2021. Dynamic Graph Construction for Improving Diversity of Recommendation. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3460231.3478845

1 INTRODUCTION

Diversified recommendation helps users to broaden interest and improve the online shopping experience. This topic has gained tremendous attention in recommender systems field [10]. For example, Maximal Marginal Relevance (MMR) [2] and Determinantal Point Process (DPP) [3] are widely adopted in numerous recommender systems. MMR is a pioneer work for diversification and it re-ranks the items based on greedy algorithms to minimize redundancy [18]. And DPP-based methods are considered to be state-of-the-art methods. DPP maintains a kernel matrix based on the users and items embeddings in a unified space and selects a set of diverse items for each user. Various efficient embedding techniques can be applied to construct the kernel matrix. With the development of graph embedding, the user-item interaction can be constructed as a graph and the graph structure allows algorithms to exploit the complex relationship between users and items. The method DivKG [5] combines the graph embedding method and DPP to enhance diversity. DivKG first adopts the TransE [1] method to get users and items embeddings by optimizing a margin-based loss and then adopts the acquired embeddings to construct the kernel matrix of

DPP to select the diverse items. The DPP-based methods such as DivKG have achieved the trade-off between relevance and diversity, but they still have some limitations and face some challenges.

First, both the MMR-based and DPP-based approaches are not end-to-end, where the optimization of the embeddings and diversity are separated into two stages. This property inevitably affects the ability to obtain the global optimal solution. Second, real-world datasets are usually imbalanced where most users only interact with a small portion of items, making it difficult for capturing users' potential interests. In this paper, we overcome the aforementioned weaknesses of the above methods with two basic observations: i) Since the user-item interaction can be naturally built as a graph, adopting the Graph Neural Networks (GNNs) to learn users and items embeddings is an reasonable choice. ii) At the same time, the graph structure can be enriched dynamically, which equips the users with more item neighbors. With the affected of the diverse items, the embedding and diversity can be optimized in an end-to-end manner.

Based on these observations, we propose an end-to-end diversified recommendation framework, namely, Dynamic Diversified Graph (DDGraph). DDGraph enriches the graph structure dynamically by selecting a set of diverse items for each user and building links between the items and user. To select the set of diverse items, we design a Quantile Progressive Candidate Selection (QPCS) algorithm. QPCS divides the latent space into multiple regions using q-quantiles and selects items in different regions to improve the diversity. To summarize, our major contributions are as follows:

• To our best knowledge, DDGraph is the first framework to use the dynamic graph construction with the diversified selection operator, and the framework makes the diversified recommendation in an end-to-end manner.

• The proposed diversified selection operator QPCS which is motivated by the idea of latent space division not only improves recommendation diversity but also has higher computational efficiency than MMR and DPP.

• We conduct extensive experiments on MovieLens-100K dataset to verify the effectiveness of the combination of dynamic graph learning and quantile progressive candidate selection inference for diversified recommendation tasks.

2 RELATED WORK

2.1 Diversified Recommendation

Diversity plays an important role in making the recommendation results more interesting and meaningful for the benefit of both users and business providers. As a classical diversified recommendation method, MMR [2] uses the notion of marginal relevance to combine the relevance and diversity with a trade-off parameter. In recent years, many DPP-based methods have been proposed to improve the diversity of the recommended results. For example, Liu [11] propose a novel diversified recommendation model (DC^2B), which employs DPP in the recommended procedure to promote diversity of the recommendation results. More recently, graph-based algorithms have shown great power in the diversified recommendation. Zheng et al. [18] propose DGCN which performs rebalanced neighbor discovering, category-boosted negative sampling, and adversarial learning on top of GNNs.

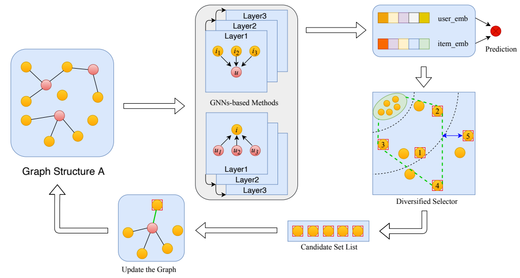

Figure 1: The overall flowchart of the proposed DDGraph.

2.2 Graph Neural Networks in Recommender System

In recent years, GNNs-based algorithms have shown a light on the recommender systems [17] [13] [15]. The original GNN algorithm proposed by Kipf et al. [8] [9] creates a classical neighbor information aggregation method and has been demonstrated to be powerful on graph data. To build the social recommender systems based on GNNs, Fan et al. [4] present a graph neural network framework (GraphRec) in which they provide a principled approach to jointly capture interactions and opinions in the user-item graph. Also motivated by the strength of graph convolution network, recent efforts like NGCF [14] and KCGN [15] adapt GNNs to the user-item interaction graph and extend the advantages of GNNs to the field of recommender systems. However, directly applying the GNNs to the field of the recommender system will not only inevitably increases the training difficulty but also cause a lot of useless operations. Aim this problem, He et al. [7] proposed LightGCN, which abandons the feature transformation and nonlinear activation in NGCF and is more concise and appropriate for the recommendation.

2.3 The Novelty of Our Method

Our model learns embedding via the GNNs-based methods and the embeddings are equipped with diverse information. The difference between the DDGraph and the other GNNs-based models is that the graph structure in our method is dynamically updated based on the selector operator QPCS. Furthermore, compared with DPP-based and MMR-based methods, our framework is an end-to-end model instead of two-stage.

3 METHODOLOGY

We now present the proposed DDGraph method whose flow is displayed in Fig. 1. It contains three key components: (i) Graph embedding for the users and items representations, which builds the user-item interactions into a graph structure and applies the graph neural network to get the users and items representations. (ii) Diversified select operator, which selects the diversified items for each user based on the users and items representations. (iii) Graph updating framework, which updating the graph structure according to the results of the diversified select operator.

3.1 Graph Embedding for User and Item Representations

Let the user-item interaction matrix be $\mathbf{R} \in \mathbb{R}^{m \times n}$ where m and n denote the number of users and items, respectively. The user set $\mathbf{U}$ is represented as $\mathbf{U} = \{u_1, u_2, \ldots, u_m\}$ and item set $\mathbf{I}$ is represented as $\mathbf{I} = \{i_1, i_2, \ldots, i_n\}$. Each entry R_{ui} is 1 if u has interacted with item i otherwise 0. Suppose the embedding matrix be $\mathbf{E} \in \mathbb{R}^{(m+n) \times d}$ and $\mathbf{E}^T = \{\mathbf{e}_{u1}, \ldots, \mathbf{e}_{um}, \mathbf{e}_{i1}, \ldots, \mathbf{e}_{in}\}$, where d is the embedding size, $\mathbf{e}_u$ is the user embedding and $\mathbf{e}_i$ is the item embedding. $(\cdot)^T$ denotes the transpose of the argument vector or matrix. The adjacency matrix of the user-item graph can be obtained as:

$$\mathbf{A} = \begin{pmatrix} \mathbf{0} & \mathbf{R} \\ \mathbf{R}^T & \mathbf{0} \end{pmatrix}. \tag{1}$$

Then the calculation of the embeddings is:

$$\mathbf{E} = AGG(\hat{\mathbf{A}}) = AGG(\mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}), \tag{2}$$

where the AGG is an aggregation function—the core of graph convolution that aggregates the information of node neighbors into this node. $\hat{\mathbf{A}}$ is the symmetrically normalized matrix in GNNs, and degree matrix $\mathbf{D}$ is a $(m+n) \times (m+n)$ diagonal matrix, in which each entry D_{ii} denotes the number of nonzero entries in the i-th row vector of the adjacency matrix $\mathbf{A}$. The method based on GNNs used here plays a role in calculating the embedding, and one can employ various embedding methods to infer the latent space.

3.2 Diversified Select Operator

Based on the graph and the latent space, we can dynamically update the graph structure by selecting several diversified neighbors for each user. For user u, the k_u interacted items are denoted as a set P_u, and the $n - k_u$ items that the user didn't interacted with are denoted as a candidate set C_u, thus $I = P_u \cup C_u$.

The diversified neighbors set N_u is the set of items selected by the candidate selection operator from C_u, and is initialized as an empty set. We can use DPP or similarity-based method (Sim) as candidate selection operators. In this work, we design a novel QPCS operator based on the latent space division idea.

Denote the distance between an item in C_u and the set $P_u \cup N_u$ as $Dist(i_{C_u}, P_u \cup N_u)$ for item $i_{C_u} \in C_u$. The distance metric can be defined according to the property of the latent space. In our work we define $Dist(i_{C_u}, P_u \cup N_u) = \min \| i_{C_u} - i_{P_u \cup N_u} | i_{P_u \cup N_u} \in P_u \cup N_u \|_2$. QPCS divides C_u into q sub-regions $\{C_u^1, C_u^2, \cdots, C_u^q\}$ based on the distance defined above, so that for any item belonging to C_u^j, its distance is smaller than any items belonging to C_u^{j+1}.

QPCS progressively selects items from C_u^1 to C_u^q. For each subregion, QPCS greedily select items with the highest score $r_{i,u}$ and adds the items to the set N_u. The score $r_{i,u}$ function is a trade-off between relevance and diversity:

$$r_{i,u} = \lambda * Rel(i,u) + (1-\lambda) * Dist(i, P_u \cup N_u) \quad \text{where} \quad i \in C_u^j, \tag{3}$$

where $Rel(i,u)$ represents the relevance score of item i to user u, $Dist(i, P_u \cup N_u)$ represents the diversity score of item i, and λ is a parameter. The distances between any of the items and N_u will be updated each time an item is added to N_u. The latent space division using q-quantiles and the timely update of N_u makes the QPCS not only consider the diversity between candidate items and P_u but also consider the diversity between the newly added items in N_u. The complete algorithm is summarized in Algorithm 1.

Algorithm 1 Quantile Progressive Candidate Selection Method (QPCS)

Input: User-interacted set P_u, candidate set C_u, score vector $\mathbf{v}$ and *stopping criteria*.
Initialize: Divide the C_u into q parts using q-quantiles and initialize the diversified neighbors set N_u as {}.

```
While stopping criteria not satisfied do:
    for each part in C_u:
        for item in the part of C_u
            calculate the score r_{i,u} with Eq. (3)
        find the item i_h with highest score
        N_u adds the item i_h
        C_u remove item i_h
```

Output: The diversified neighbors set N_u.

3.3 The Updating of the Graph Construction

The graph structure can be updated based on the diversified neighbors set N_u and the original user-item interaction matrix $\mathbf{R}$. Let the diversified user-item interaction matrix be $\mathbf{R}^{div} \in \mathbb{R}^{m \times n}$ and each entry R_{ui}^{div} is 1 if item i in the diversified neighbors set N_u otherwise 0. With exploring the new neighbors for users in N_u, the adjacency matrix becomes $\mathbf{A}^{div}$.

$$\mathbf{A}^{div} = \begin{pmatrix} \mathbf{0} & \mathbf{R} + \mathbf{R}^{div} \\ (\mathbf{R} + \mathbf{R}^{div})^T & \mathbf{0} \end{pmatrix}. \tag{4}$$

Then the calculation of the embeddings is:

$$\mathbf{E}^{div} = AGG(\hat{\mathbf{A}}^{div}) = AGG((\mathbf{D}^{div})^{-\frac{1}{2}} \mathbf{A}^{div} (\mathbf{D}^{div})^{-\frac{1}{2}}). \tag{5}$$

4 EXPERIMENTS

4.1 Dataset

The MovieLens-100K [6] dataset is a benchmark dataset containing 100,000 user ratings ranging from score 1 to 5 from 943 users on 1,682 movies. We follow the traditional idea to binarize explicit rating data by keeping the ratings of four or higher. At the same time, the leave-one-out method is used to divide the data into the training set and test set.

4.2 Compared Methods

To verify the effectiveness of our method, we compare it with following methods.

- **LightGCN** [7]: A state-of-the-art GCN-based model. LightGCN simplifies the design of GCN by discarding the feature transformation and nonlinear activation to make it more concise and appropriate for recommendation.
- **BPRMF** [12]: A state-of-the-art matrix factorization model optimized by a pairwise ranking loss to learn from implicit feedback. The proposed Bayesian Personalized Ranking (BPR) loss is also used in LightGCN and our method.

Table 1: Relevance and Diversity Comparison on MovieLens-100K.

Model	hits@5	hits@10	hits@20	ndcg@5	ndcg@10	ndcg@20	ILAD	ILMD
BPRMF	15.80%	23.75%	35.73%	10.65%	13.23%	16.25%	24.15%	4.08%
BPRMF-DPP	11.13%	19.72%	30.96%	7.20%	9.90%	12.74%	30.92%	5.31%
LightGCN	17.92%	**30.01%**	**45.28%**	11.99%	15.92%	19.75%	36.49%	8.15%
LightGCN-DPP	17.60%	29.48%	42.20%	11.87%	15.66%	18.88%	39.72%	10.25%
DDGraph(Sim)	18.87%	29.79%	45.06%	12.50%	15.99%	19.83%	41.44%	9.37%
DDGraph(DPP)	18.13%	29.05%	44.64%	12.00%	15.54%	19.47%	38.48%	8.84%
DDGraph(QPCS)	**19.30%**	29.79%	44.64%	**13.06%**	**16.45%**	**20.22%**	**42.07%**	**10.96%**

Table 2: The time cost of calculating the recommendation list (ms).

LightGCN-DPP	BPRMF-DPP	DDGraph(Smi)	DDGraph(DPP)	DDGraph(QPCS)
14.04	21.18	0.87	0.87	0.89

Table 3: The performance of QPCS as the post-process.

	LightGCN-Embedding			BPRMF-Embedding		
	MMR	DPP	QPCS	MMR	DPP	QPCS
ILAD	51.70%	54.27%	**73.41%**	28.39%	45.30%	**82.45%**
ILMD	**16.36%**	15.86%	14.52%	5.82%	**7.92%**	4.69%

• **DPP** [3]: DPP is an elegant probabilistic model that is widely used for diversified recommendation. We use Fast Greedy MAP Inference algorithm and construct kernel matrix the same way in [3].

• **MMR** [2]: MMR is a classic diversified recommendation algorithm. It defines relevance and diversity by independent metrics and chooses items one by one by maximizing the objective of marginal relevance in a greedy manner.

• **DDGraph(DPP)**: The diversified neighbors set N_u is selected using DPP.

• **DDGraph(Sim)**: The diversified neighbors set N_u is selected basing on the similarities between embeddings.

4.3 Experimental Settings

To reduce the experiment workload and keep the comparison fair, we follow the same hyper-parameter settings reported by the authors for the baselines except the embedding dimension,which is set to 100 for all methods in our experiments. The size of the diversified neighbors set N_u is 10, and the number of the latent space part q is 4. Besides, we update the graph structure every 50 epochs during the training.

To evaluate the relevance and diversity of our framework, we adopt widely-used evaluation protocols hits@K and ndcg@K [16] for relevance, Intra-List Average Distance (ILAD) and Intra-List Minimal Distance (ILMD) [3] for diversity. We report the experimental results from the following three aspects: relevance and diversity comparison between DDGraph and baselines, diversity comparison between QPCS and other post-processing methods, and computational efficiency.

4.4 Task 1: Relevance and Diversity Comparison

• **Relevance and Diversity Comparison.** The comparison results are summarized in Table 1, we can conclude that:

i) LightGCN/LightGCN-DPP is better than BPRMF/BPRMF-DPP on both relevance and diversity. It proves that graph neural network has better support for capturing the complex interaction information.

ii) DDGraph outperforms all the other baselines in both ILAD and ILMD. Compared to LightGCN-DPP, DDGraph(QPCS) is 2.35% higher on ILAD and 0.71% higher on ILMD. The results support that the dynamic graph construction is capable of incorporating complex non-interaction information which is very beneficial for diversity. Besides, DDGraph(QPCS) is superior to LightGCN-DPP in hits and ndcg, indicating that dynamic graph construction is also helpful for relevance.

iii) DDGraph(QPCS) is superior to DDGraph(Sim) and DDGraph(DPP) where the only difference among them is the candidate selector operator. Therefore, it is evident that the QPCS is more powerful for selecting diverse items.

• **Computational Efficiency.** The average time cost of calculating the diversified items list for one user is shown in Table 2 where we report the average results of 10 experiments. By brings diversity information into the graph embeddings via QPCS, our method can be trained end-to-end where the final recommendation list with the diverse information need no other post-process. Our experiments for computational efficiency are performed on a MacBook Pro machine with 2.60-GHz, Core i7 CPU, and 16-GB RAM.

Table 4: The time cost of calculating the recommendation list (ms).

MMR-LightGCN	DPP-LightGCN	QPCS-LightGCN	MMR-BPRMF	DPP-BPRMF	QPCS-BPRMF
1.09	12.89	**0.96**	1.14	20.78	**0.86**

4.5 Task 2: QPCS as the post-processing

• **Diversity Comparison.** To verify the superiority of QPCS compared to DPP and MMR on diversified recommendation, we design ablation experiments based on LightGCN and BPRMF. Based on the embeddings generated by the LightGCN and BPRMF, the final recommendation list is calculated by MMR/DPP/QPCS. Since the results are a trade-off between diversity and relevance, we adjust the parameter λ in Eq. (3) to make the performance on the relevance of all methods is roughly the same.

The experiment results are shown in Table 3, from which we can conclude that:

i) Based on the embedding of LightGCN, QPCS is 21.71% higher than MMR and 19.14% higher than DPP on ILAD. As for the embedding of BPRMF, QPCD is 54.06% higher than MMR and 37.15% higher than DPP on ILAD.

ii) Thanks to the division of the latent space, items are evenly scattered among different sub-regions, which leads to the tremendous increase in ILAD. Since the distance between items belonging to the same sub-region is restricted by the size of the sub-region, there is a gentle decrease on ILMD. However, real-world users care more about the average diversity of items rather than the most unfamiliar one, thus optimizing ILAD is more practical than optimizing ILMD.

• **Computational Efficiency.**

When the embeddings are generated by LightGCN and BPRMF, MMR/DPP/QPCS are used to calculate the final recommendation list. The average time cost for one user is shown in Table 4. The results in Table 4 show the advantages of our method over MMR and DPP on time cost.

5 CONCLUSION

In this paper, we develop a Dynamic Diversified Graph framework, namely DDGraph, which constructs a dynamic graph structure to explore the user-item non-interactive information for diversified recommendation in an end-to-end manner. DDGraph builds a graph structure based on user-item interaction information where the graph structure can be dynamically updated by selecting a set of diverse items and building links to them for each user. To select the diverse items, we design a QPCS which divides the latent space using q-quantiles and greedily selects the candidate with the highest score progressively. DDGraph learns embeddings via the GNNs-based methods and the dynamically added diverse items bring diverse information into the embeddings. Extensive offline experiments on the benchmark dataset illustrate the effectiveness and superiority of the proposed method.

REFERENCES

[1] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[2] Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 335–336.

[3] Laming Chen, Guoxin Zhang, and Hanning Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. 5627–5638.

[4] 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.

[5] Lu Gan, Diana Nurbakova, Léa Laporte, and Sylvie Calabretto. 2020. Enhancing recommendation diversity using determinantal point processes on knowledge graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2001–2004.

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

[7] 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.

[8] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[9] Thomas N Kipf and Max Welling. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308* (2016).

[10] Matevž Kunaver and Tomaž Požrl. 2017. Diversity in recommender systems–A survey. *Knowledge-based systems* 123 (2017), 154–162.

[11] Yong Liu, Yingtai Xiao, Qiong Wu, Chunyan Miao, Juyong Zhang, Binqiang Zhao, and Haihong Tang. 2020. Diversified interactive recommendation with implicit feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 4932–4939.

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

[13] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 950–958.

[14] 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.

[15] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.

[16] Jheng-Hong Yang, Chih-Ming Chen, Chuan-Ju Wang, and Ming-Feng Tsai. 2018. HOP-rec: high-order proximity for implicit recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*. 140–144.

[17] 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.

[18] Yu Zheng, Chen Gao, Liang Chen, Depeng Jin, and Yong Li. 2021. DGCN: Diversified Recommendation with Graph Convolutional Networks. In *Proceedings of the Web Conference 2021*. 401–412.