Motif-aware Sequential Recommendation

Zeyu Cui[1,2], Yinjiang Cai[2,3], Shu Wu[1,2,*], Xibo Ma[1,2], Liang Wang[1,2]
[1]Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Science
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]CBSR&NLPR, Institute of Automation, Chinese Academy of Sciences
Beijing, China
{zeyu.cui,yinjiang.cai,shu.wu,xibo.ma,wangliang}@nlpr.ia.ac.cn

ABSTRACT

Sequential recommendation is intended to model the dynamic behavior regularity through users' behavior sequences. Recently, various deep learning techniques are applied to model the relation of items in the sequences. Despite their effectiveness, we argue that the aforementioned methods only consider the macro-structure of the behavior sequence, but neglect the micro-structure in the sequence which is important to sequential recommendation. To address the above limitation, we propose a novel model called Motif-aware Sequential Recommendation (MoSeR), which captures the motifs hidden in behavior sequences to model the micro-structure features. MoSeR extracts the motifs that contain both the last behavior and the target item. These motifs reflect the topological relations among local items in the form of directed graphs. Thus our method can make a more accurate prediction with the awareness of the inherent patterns between local items. Extensive experiments on three benchmark datasets demonstrate that our model outperforms the state-of-the-art sequential recommendation models.

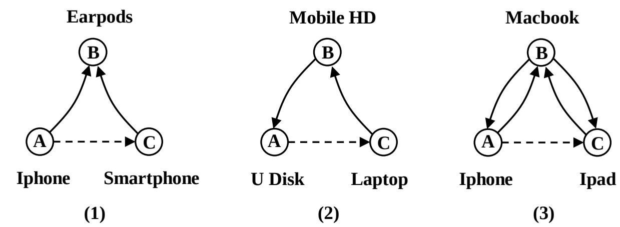


Figure 1: Illustration of different motifs in sequential recommendation. Each solid arrow represents a common global purchase order between items. The dotted arrows denote the behaviors we need to predict. Three motifs present the collision structure, unidirectional dependency and bidirectional dependency relations respectively.

CCS CONCEPTS

• **Applied computing → Online shopping**; • **Information systems → Recommender systems**.

KEYWORDS

Sequential recommendation, Graph structure, Motif

ACM Reference Format:
Zeyu Cui[1,2], Yinjiang Cai[2,3], Shu Wu[1,2,*], Xibo Ma[1,2], Liang Wang[1,2]. 2021. Motif-aware Sequential Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21), July 11–15, 2021, Virtual Event, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3404835.3463115

1 INTRODUCTION

With the explosive growth of information on the Internet, recommendation systems have been widely used in many online services such as e-commerce, advertising and information retrieval to alleviate information overload. Sequential recommendation is one of the fundamental tasks in recommendation systems which aims to model the behavior regularity through users' behavior sequences.

Various methods have been proposed for the sequential recommendation. Traditional methods [1, 11] model the relation from an item to the next item. Recently, deep learning methods such as GRU4Rec, Caser [3, 4, 7, 13], employ recurrent neural network or convolution neural network to learn the sequential regularity of user-item interactions. Nevertheless, these methods fail to capture some complex sequential characteristics of behaviour, such as long term relation, periodical relation. With the widespread use of attention mechanism, many models [6, 12] are proposed to introduce attention to distinguish the importance of items in sequences. These attention-based methods can provide a more accurate picture of the entire sequence. Besides the sequential structure of models, graph-based methods [15, 16] are proposed to capture complex transitions of items which are difficult to be revealed by the conventional sequential methods. They combine all the sequences into an item graph to learn the global relations of item transition features and achieve satisfactory results.

The aforementioned methods focus on the macro-structure of the whole sequence. However, they pay little attention to the micro-structure hidden in the sequence. Recent research [5, 9] on the motif of the dynamic graph shows that micro-structure is highly relevant to the future linkage. Such micro-structure is also common and important in recommendation systems. Taking Figure 1 as examples, solid arrows demonstrate users' frequent interaction patterns from one item to another. Different types of motifs explicitly reflect different micro-structure. Motif (1) shows collision relation, as one

*To whom correspondence should be addressed.

SIGIR '21, July 11–15, 2021, Virtual Event, Canada

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463115

may not buy another cell phone just after an Iphone. Motif (2) represents unidirectional dependency, since a user probably needs a Mobile HD and U Disk after having a Laptop, while seldom buying a Laptop after a U Disk. Motif (3) is a bidirectional dependency relation, where a user has a high probability to buy others after having any one in the motif. These micro-structure features explicitly model the probability of next items through topological relations between local items.

To address the above limitation, we propose to use motif to model micro-structure features for the sequential recommendation model. Specifically, we summarize all sequences of training set into a directed item graph and extract motifs that contain the last behavior and the target item from the directed graph. For the sake of complexity and sparsity, we concentrate on the simplest and fundamental motif, *triad*. The motif features are calculated by edge weights in extracted motifs. Finally, we combine the motif features, representation of user sequence and the target item representation as the input to predict the probability of the target item at the next time step. We conduct extensive experiments on three benchmark datasets. The experimental results demonstrate the effectiveness of the proposed method over the state-of-the-arts. The main contributions of this work are as follows,

- We design a micro-structure motif feature hidden in users' behavior sequences which plays an important role in sequential recommendation.
- We propose a motif-aware sequential recommendation model called MoSeR, which takes motif into account in the predicting stage.
- Experiments on three public datasets show the effectiveness of our model in capturing micro-structure features. The code will be released after publication.

2 PROPOSED METHODS

In this section, we present a novel method called MoSeR. An item graph is constructed from user behavior at the beginning in Section 2.2. Then, we capture the motif feature based on $\mathcal{G}$ in Section 2.3. We introduce a sequential recommendation model based on Transformer [6, 14] to combine the motif feature with item sequence in Section 2.4.

2.1 Problem Formulation

Let Q denote the item set and $|Q|$ denote the number of items. In sequential recommendation task, the behavior sequence of a user u is denoted as $s^u = [s^u_1, s^u_2, ..., s^u_t, ..., s^u_T]$, where T is the length of sequence, s^u_t represents the item index in Q that user u interacts with at time step t. The recommendation task is to predict the potential item s^u_{T+1} at the next time step.

2.2 Item Graph Construction

To capture the structural information among items, we summarize all the sequence relation of items into a directed graph. Similar to the previous graph based sequential recommendation method [15], we construct a weighted directed graph $\mathcal{G}$ by the behaviour order of items. Each node of $\mathcal{G}$ represents an item. Each edge denotes a directed transition from one item to another. The weight of an edge (i, j) is calculated by the frequency of the transition relation

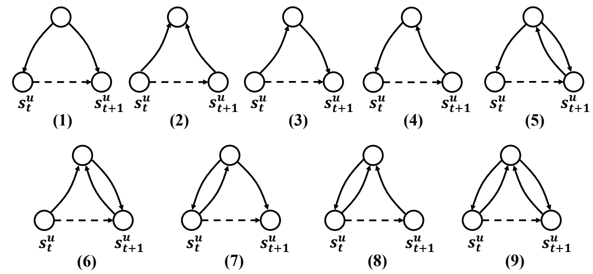


Figure 2: Nine types of open triad motifs.

occurs on the training dataset, which is written as,

$$A(i,j) = \frac{\sum_u p(s^u_t = i, s^u_{t+1} = j)}{\sum_u p(s^u_t = i)}, \tag{1}$$

where $p(s^u_t = i, s^u_{t+1} = j)$ denotes the frequency of u interacting with item i and then j at any time t. When u never interacts with i after j, it equals 0. $p(s^u_t = i)$ represents the frequency of u interacting with i. Based on the item graph, we could extract the motif features for sequential recommendation.

2.3 Motif Feature Extraction

From $\mathcal{G}$, we could discover the motif patterns of items. For simplicity, we only discuss the motifs containing three nodes, i.e., triad motifs. Predicting the next item s^u_{t+1} could be considered as learning the probability of a future links from s^u_t to s^u_{t+1}. In that case, the most related motifs are the open triad motifs containing s^u_t and s^u_{t+1}, which are shown in Figure 2. There are totally nine types of motifs. The (k)-type open triad motif set contains s^u_t and s^u_{t+1} could be written as,

$$H_{\Delta_k}(s^u_t, s^u_{t+1}) = [(s^u_t, m_1, s^u_{t+1}), (s^u_t, m_2, s^u_{t+1}), ...], \tag{2}$$

where $[m_1, m_2, ...]$ are the middle node indexes of the triad which could be any item in the training set. For each triad (s^u_t, m, s^u_{t+1}), we define a function $\sigma(s^u_t, m, s^u_{t+1})$ that represents the importance of a motif by the edge weight as,

$$\sigma(s^u_t, m, s^u_{t+1}) = A(s^u_t, m) + A(m, s^u_t) + A(s^u_{t+1}, m) + A(m, s^u_{t+1}). \tag{3}$$

Note that the edge weight is zero when it does not exist. The function $\sigma(\cdot)$ could be designed as other formulations. We use adding function for simplification. Finally, we extract the motif feature $\mathbf{X}(s^u_t, s^u_{t+1}) \in \mathbb{R}^9$ for each item pair. Each dimension of $\mathbf{X}(s^u_t, s^u_{t+1})$ corresponds to one type of motif pattern,

$$\mathbf{X}_k(s^u_t, s^u_{t+1}) = \sum_{(s^u_t, m, s^u_{t+1}) \in H_{\Delta_k}(s^u_t, s^u_{t+1})} \sigma(s^u_t, m, s^u_{t+1}). \tag{4}$$

In practice, processing all the motifs brings much computational cost. Thus we fix the maximum number of candidate triads as M_{max} in this paper, select candidate triads randomly and discuss its influence in section 3.3.

2.4 MoSeR Framework

In this section, we apply the extracted motif feature for sequential recommendation. Similar to the practice in SASRec [6], we randomly initialize item representation as $\mathbf{Q} \in \mathbb{R}^{|Q| \times d}$. Then, we model the user behavior sequence by stacking transformer blocks. The concatenation of the sequence representation and motif features is the input of the prediction layer and output is considered as the user's preference on the next item.

2.4.1 Position-aware transformer blocks. Given the user behavior sequence $s^u = [s_1^u, s_2^u, ..., s_t^u, ...s_T^u]$. After injecting the learnable position embedding $\mathbf{P} \in \mathbb{R}^{T \times d}$, the input matrix $\mathbf{S}$ is written as,

$$\mathbf{S} = [\mathbf{Q}_{s_1^u} + \mathbf{P}_1, \mathbf{Q}_{s_2^u} + \mathbf{P}_2, ..., \mathbf{Q}_{s_T^u} + \mathbf{P}_T], \tag{5}$$

where $\mathbf{Q}_{s_t^u}$ is the representation of item s_t^u.

The common transformer block has the input of query, key and value. Here, we use $\mathbf{S}$ as the input, and convert it into query, key and value by three learnable projection matrices. The transformer block is defined as following,

$$\text{Attention}(\mathbf{S}) = \text{softmax}\left(\frac{\mathbf{SW}^1(\mathbf{SW}^2)^T}{\sqrt{d}}\right)\mathbf{SW}^3, \tag{6}$$

where $\mathbf{W}^1, \mathbf{W}^2, \mathbf{W}^3 \in \mathbb{R}^{d \times d}$ are the projection matrices. The block uses the whole behavior history $\mathbf{S}$ to generate the attention weight for items at each time step. We denoted $\mathbf{S}^{(b)}$ as the output after (b)-th layers of transformer blocks. The final representation of user preference $\mathbf{F}$ on the next item is obtained through a 2-layer fully-connected networks (FCN), which is written as,

$$\begin{aligned} \mathbf{S}^{(b)} &= \text{Attention}(\mathbf{S}^{(b-1)}), \\ \mathbf{F} &= \text{FCN}_{\theta_1}(\mathbf{S}^{(1)} + \mathbf{S}^{(b)}), \end{aligned} \tag{7}$$

where θ_1 is the learnable parameters of FCN. We add the first layer $\mathbf{S}^{(1)}$ as a residual connection which makes the training of lower layer representation better.

2.4.2 Motif-aware next item prediction. The final representation $\mathbf{F}$ denotes the user's preference on the next item, where $\mathbf{F}_t$ is the preference at time $t+1$. To predict the rate $r_{u,i,t}$ of item i that whether user u interacts with it at the next time step $t+1$, we aggregate the representation $\mathbf{F}_t$, $\mathbf{Q}_i$ and the motif feature $\mathbf{X}(s_t^u, i)$ and output a scalar through a 2-layer FCN,

$$r_{u,i,t} = \text{FCN}_{\theta_2}(\text{concatenate}(\mathbf{F}_t, \mathbf{Q}_i, \mathbf{X}(s_t^u, i))), \tag{8}$$

where θ_2 is the trainable parameters of the FCN.

2.5 Training Strategy

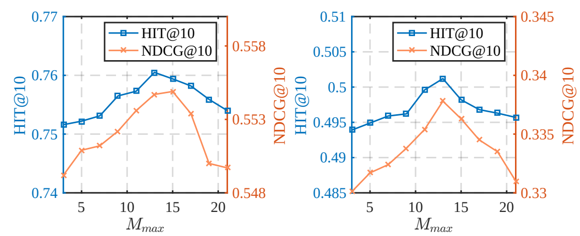
In the training stage, we fix each user sequence to a certain length. We cut the early records when the sequence exceeds the maximum length, and pad default zero vectors at the beginning when the sequence is not long enough. We adopt the binary cross entropy loss as the objective function,

$$J = -\sum_u \sum_t^T \left[\log(f(r_{u,s_{t+1}^u,t})) + \sum_{j \notin s^u} \log(1 - f(r_{u,j,t}))\right], \tag{9}$$

where $f(\cdot)$ is the sigmoid function written as, $f(x) = 1/(1 + \exp(-x))$. In each epoch, we randomly generate one negative item for each time step in each sequence. More implementation details are described in the experiment Section 3.1.

Table 1: Statistics of Datasets.

Dataset	#Users	#Items	Avg.length	#Actions
Games	31,013	23,715	9.3	0.3M
Beauty	52,024	57,289	7.6	0.4M
MovieLens-1M	6,040	3,416	163.5	1.0M



(a) Games (b) Beauty

Figure 3: Performance of MoSeR with different maximum candidate motif M_{max}.

3 EXPERIMENTS

In this section, we first describe the experimental settings. We then report results by answering the above research questions in turn.

3.1 Experiment Setup

Datasets. Following [1, 2, 11], we conduct experiments on the following three real-world datasets, whose statistics can be found in Table 1. **Games, Beauty** [8] are two subsets of Amazon dataset[1]. It is widely used as a benchmark dataset in the recommendation, which contains product reviews and metadata from Amazon. **MovieLens**[2] is a movie rating dataset which is widely used for evaluating recommendation algorithms. We use the well-established version (MovieLens-1M) that includes 1 million user ratings. We follow the same preprocessing procedure as [1, 2, 11]. We discard users and items with fewer than 5 related actions, and adopt the leave-one-out protocol for evaluation. For each user, we use the last behavior for testing, the second to last for validation and the others for training. During testing, the input sequences contain training actions and the validation action. Data statistics are shown in Table 1 in detail.

Compared Methods. We compare MoSeR with eight methods. Two static methods include PopRec (most popular recommendation) and Bayes Personalized Ranking (BPR) [10]. Three traditional sequential methods include Factorized Markov Chains (FMC), Factorizing Personalized Markov Chains (FPMC) [11] and Translation-based Recommendation (TransRec) [1]. Three sequential neural network methods include GRU4Rec$^+$ [3, 4], Convolutional Sequence Embeddings (Caser) [13] and SASRec [6].

Settings. Following the experiment setting of previous work [6], we consider latent dimensions d from $[10, 20, 30, 40, 50]$ for all methods except PopRec. For traditional methods, the l_2 regularizer is chosen from $[0.0001, 0.001, 0.01, 0.1, 1]$. In our model, the length of behaviors is set as 200 for Movielens and 50 for others. Two

[1] http://jmcauley.ucsd.edu/data/amazon/links.html
[2] https://grouplens.org/datasets/movielens/1m/

Table 2: The performance of sequential recommendation on three datasets.

		PopRec	BPR	FMC	FPMC	TransRec	GRU4Rec+	Caser	SASRec	MoSeR
Games	Hit@10	0.4724	0.4853	0.6358	0.6802	0.6838	0.6599	0.5282	<u>0.7410</u>	**0.7605**
	NDCG@10	0.2779	0.2875	0.4456	0.468	0.4557	0.4759	0.3214	<u>0.5360</u>	**0.5536**
Beauty	Hit@10	0.4003	0.3775	0.3771	0.431	0.4607	0.3949	0.4264	<u>0.4854</u>	**0.5012**
	NDCG@10	0.2277	0.2183	0.2477	0.2891	0.302	0.2556	0.2547	<u>0.3219</u>	**0.3361**
Movielens	Hit@10	0.4329	0.5781	0.6986	0.7599	0.6413	0.7501	0.7886	<u>0.8245</u>	**0.8306**
	NDCG@10	0.2377	0.3287	0.4676	0.5176	0.3969	0.5513	0.5538	<u>0.5905</u>	**0.6004**

stacked transformer blocks are used to model sequences. We tune hyper-parameters using the validation set, and terminate training if validation performance does not improve for 20 epochs. We use HIT@10 and NDCG@10 to evaluate the performance of our methods.

3.2 Model Comparison

Table 2 shows the performance of all compared methods on three datasets. Traditional static methods, PopRec and BPR get the worst results on all the three datasets, since they ignore the sequential information of user behaviors. As traditional sequential methods FMC, FPMC and TransRec focus on explicitly modeling the sequential relations between item pairs, they can even get better results than some neural sequential methods on some datasets. It indicates the importance of micro relations hidden in the sequence, especially on sparse datasets such as Games and Beauty. SASRec uses attention mechanism to adaptively attend items within different ranges on different datasets and achieves better performance than other baseline methods. However, it still has difficulty in capturing the micro-structure information of items. For this reason, our model MoSeR achieves the best results on all the three datasets. Especially on Beauty, it achieves an improvement of 3.25% with respect to HIT@10 and 4.41% with respect to NDCG@10. Compared with SASRec, MoSeR has the similar sequential neural networks structure to catch the sequential patterns. Thus, the improvements mainly come from motif features.

3.3 Influence of M_{max}

Calculating all the motifs of each item pair brings too much computation cost. Here, we discuss how the maximum number of candidate motifs M_{max} selected for a pair of items may influence the performance. Figure 3 shows the results with different M_{max}. Generally, the performance of MoSeR increases with the increasing of M_{max}, and decreases when M_{max} goes too large. It is reasonable since the motif feature may not be sufficient when the candidate motifs are few. However, when the candidate motif set is too large, some rare but important types of motifs may lose their effect when there are too much useless candidate motifs. On both datasets the best performance is achieved when M_{max} equal to 13.

3.4 Influence of Different Motif Patterns

We present the feature characteristics of each motif pattern in Figure 4. The x-axis denotes the 9 motif patterns plotted in Figure 2. The orange bar denotes the average motif feature for each type of motifs in the positive training set. The purple bar denotes the same feature in the positive testing set. Green bars indicate the

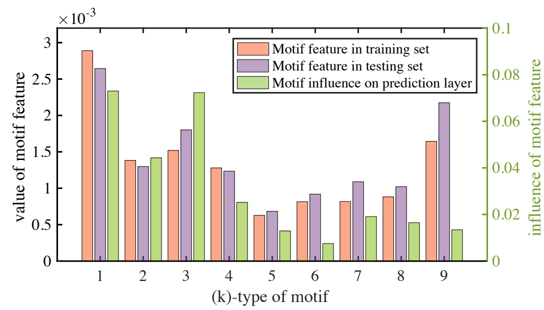


Figure 4: The visualization of motif weights and motif influence on dataset Games.

influence of motifs on the prediction layer. We obtain the influence by averaging the weight matrix of first layer in $\mathrm{FCN}_{\theta_2}(\cdot)$, since the last 9 rows of the matrix are the weight parameters for motif features. It can be seen that the feature of the testing set maintains the similar distribution as the training set, which indicates motif features perform stably on recommendation. Among the first four types of motifs, the second and fourth motifs do not appear as frequent as the other motifs, which verifies the example (1) and (2) of Figure 1 in Section 1. Thus, the influence of second and fourth motifs are lower than the others. Generally, the motifs that get the averagely high feature also have high influence on prediction layer, except the motif pattern 9. Motif 9 indicates a bidirectional relation of query and target items. Items in this motif are highly related. This relation could be easily modeled by the item representations through prediction layer without the assistance of motif features.

4 CONCLUSION

In this paper, we highlight the importance of micro-structure features in sequential recommendation and propose a novel model MoSeR. MoSeR captures micro-structure features by extracting motifs from the item graph formed by behavior sequence. The extracted motif features are used to enhance the next item prediction with transformer blocks. Experiments on three real-world datasets demonstrate that our proposed model achieves the state-of-the-art performance in sequential recommendation.

ACKNOWLEDGMENTS

This work is supported by National Key Research and Development Program (2018YFB1402605, 2018YFB1402600), National Natural Science Foundation of China (U19B2038, 61772528), Beijing National Natural Science Foundation (4182066).

REFERENCES

[1] Ruining He, Wang-Cheng Kang, and Julian McAuley. 2017. Translation-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 161–169.

[2] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining*. 191–200.

[3] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM*. 843–852.

[4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations*.

[5] Hong Huang, Zixuan Fang, Xiao Wang, Youshan Miao, and Hai Jin. 2020. Motif-preserving temporal network embedding. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*. 1237–1243.

[6] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining*. 197–206.

[7] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[8] 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.

[9] Ashwin Paranjape, Austin R Benson, and Jure Leskovec. 2017. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. 601–610.

[10] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[11] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[12] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[13] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 565–573.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[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*. 346–353.

[16] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2020. TAGNN: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1921–1924.