# Temporal Augmented Graph Neural Networks for Session-Based Recommendations

Huachi Zhou,[1] Qiaoyu Tan,[1] Xiao Huang,*[1] Kaixiong Zhou,[2] Xiaoling Wang[3]
[1]Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong
[2]Department of Computer Science and Engineering, Texas A&M University, College Station, USA
[3]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai, China
huachi.zhou@connect.polyu.hk,{qiaoytan,xiao.huang}@polyu.edu.hk,zkxiong@tamu.edu,xlwang@cs.ecnu.edu.cn

## ABSTRACT

Session-based recommendation aims to predict the next item that is most likely to be clicked by an anonymous user, based on his/her clicking sequence within one visit. It becomes an essential function of many recommender systems since it protects privacy. However, as the accumulated session records keep increasing, it becomes challenging to model the user interests since they would drift when the time span is large. Efforts have been devoted to handling dynamic user interests by modeling all historical sessions at one time or conducting offline retraining regularly. These solutions are far from practical requirements in terms of efficiency and capturing timely user interests. To this end, we propose a memory-efficient framework - TASRec. It constructs a graph for each day to model the relations among items. Thus, the same item on different days could have different neighbors, corresponding to the drifting user interests. We design a tailored graph neural network to embed this dynamic graph of items and learn temporal augmented item representations. Based on this, we leverage a sequential neural architecture to predict the next item of a given sequence. Experiments on real-world datasets demonstrate that TASRec outperforms state-of-the-art session-based recommendation methods.

## CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Retrieval models and ranking**.

## KEYWORDS

Session-based recommendations; graph neural networks; dynamic user interests

**ACM Reference Format:**
Huachi Zhou, Qiaoyu Tan, Xiao Huang, Kaixiong Zhou, Xiaoling Wang. 2021. Temporal Augmented Graph Neural Networks for Session-Based Recommendations. 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/10.1145/3404835.3463112


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

ACM ISBN 978-1-4503-8037-9/21/07…$15.00
https://doi.org/10.1145/3404835.3463112

## 1 INTRODUCTION

The session-based recommendation attracts increasing attention since it helps online users protect their privacy [3, 5]. In various real-world recommender systems, such as YouTube, Netflix, Taobao, and JD.com, many users would prefer to not log in. In these scenarios, the historical shopping or watching record of each user is not available. Conventional recommendation methods could not directly be applied to learn the user interests based on interactions. Thus, session-based recommendation has been explored. Given a sequence of items clicked by an anonymous user within one visit, the goal is to predict the next item that this user would click within the same visit [15–17].

To perform session-based recommendations, existing methods mainly focus on two tasks, i.e., effectively capturing item dependencies in one session and mining collaborative signal to enrich sole session information. First, gated recurrent units have been adopted [5] to process short sequences because its superior performance than long short term memory [3]. Wu et al. [17] utilize graph gated neural network to further explore item dependencies among distant items in each session. Second, since user behaviors involved in the intra-session are limited, item relationships among inter-sessions are incorporated to better analyze user intents. Wang et al. [16] construct a global graph over all sessions to capture potential high order signal of items. Wang et al. [15] summarize a collaborative sequence representation by extracting useful features from neighbor sessions in the outer memory.

However, in practice, numerous historical session records have been accumulated, which are informative but challenging to model. It becomes expensive to load the entire historical sessions into the memory and train the model at one time. Also, the user interests would vary since the time span is large [3, 13]. A few recent efforts have been devoted to modeling these dynamic user interests. Mi and Faltings [9] propose to employ a large memory module to learn representations for all historical sessions. When making inferences for the new coming session, they search an inverted table to find existing k nearest neighbor sessions and compute an output distribution to predict the next item. This solution could successfully utilize long-term and short-term information. It works well when the time span is small, e.g., one or two weeks. However, as the time span becomes large, it would be infeasible to hold all session records in the memory. Qiu et al. [10] propose to use a Wasserstein reservoir to hold sampled worst predicted sequences and retrain the whole model after some time. This method could easily learn the new preferences of users by retraining on new coming data. However, such model updates still could not capture new item transitions in time. To preserve useful historical information, they slow

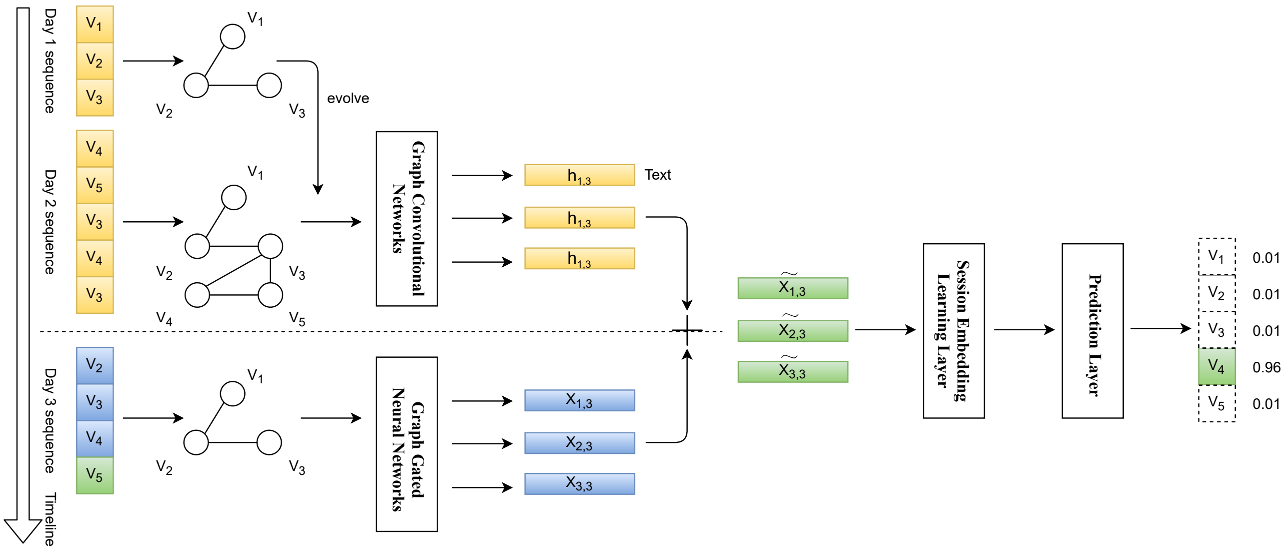


**Figure 1: TASRec employs a dynamic graph to capture the drifting user interests in session-based recommendations**

down the sampling speed as time goes by. It will overlook recent data. Thus, existing solutions can not satisfy the practical demands in efficiently and timely capturing drifting user interests.

In summary, there are three major challenges in handling session-based recommendations with long-duration and a large number of sessions. First, sessions are continuously growing. The memory could not hold all sessions. Second, the user interests might drift significantly. An appropriate model is needed to model the temporal information in the historical sessions. Third, the information contained in the new session needs to be modeled in time.

To tackle the above problems, we propose temporal augmented graph gated neural networks to extract helpful information from continuously coming sessions. In summary, our main contributions are described below.

- We propose a Temporal Augmented graph neural network for Session-based Recommendations (TASRec), to model the dynamic user interests within a long period.
- We build an efficient temporal graph that adds new edges from recent coming sessions to incorporate novel item transitions and decay old edge weights to filter outdated information. The shifting neighbors on a different day could help the model capture emerging user interests.
- We conduct experiments on real-world datasets and validate the effectiveness of TASRec.

## 2 TEMPORAL AUGMENTED RECOMMENDER SYSTEM - TASREC

**Notations.** Let $\mathcal{I}$ represent the unique item set $\{i_1, i_2..., i_m\}$ involved in all the sessions, where the cardinality size of $I$ is denoted by $|\mathcal{I}|$. A session $s$ of an anonymous user is represented by a sequence $[i_{1,t}, i_{2,t}..., i_{l,t}]$, where $i_{1,t} \in \mathcal{I}$ represents item clicked by user within session $s$ in day $t$. The sequence is ordered by clicked timestamps of the items and has length of $l$. Given such the session $s$, the objective of session-based recommender system is to predict top-N items most likely to be clicked next by the corresponding user. In this paper, we represent item-to-item interactions in two levels: session graph and temporal graph. Let $G_s = (\mathcal{V}_s, E_s)$ denote the session graph, where $\mathcal{V}_s \subset \mathcal{I}$ is item nodes in session $s$ and $E_s \in \mathbb{R}^{l \times 2l}$ is directed adjacency matrix. The matrix element $(i_{i-1,t}, i_{i,t})$ is defined by the occurrence of the edge divided by the outdegree of that edge's start node $i_{i-1,t}$ from the outgoing direction or the indegree of the end node $i_{i-1,t}$ from the incoming direction. Let $G_t = (\mathcal{V}_t, E_t)$ denotes the temporal graph utilized in day $t$, where $\mathcal{V}_t \subset \mathcal{I}$ contains items of sessions generated before day $t$ and $E_t \in \mathbb{R}^{I \times I}$ is the undirected adjacency matrix. Each element in $E_t$ is defined by the co-occurrence frequency of the corresponding two items, which will be illustrated in the following. And We present our TASRec framework in Figure 1.

### 2.1 Intra-session Modeling Layer

Following the previous common practice [16–18], we first use the standard graph gated neural networks (GGNN) on session graph $G_s$ to capture the intra-session item interactions. Given session $s$ in day $t$ and the initial embedding layer $X$, we conduct GGNN on graph $G_s$ to obtain the intra-session item embeddings: $[h_{1,t}, h_{2,t}..., h_{l,t}]$, each one corresponding to item in session sequence $[i_{1,t}, i_{2,t}..., i_{l,t}]$.

### 2.2 Temporal Modeling Layer

Considering recommendation task in session $s$ of day $t$, the temporal modeling layer targets at learning the item interactions on graph $G_t$, which stores the historical session records before day $t$. To facilitate such the interaction modeling, we propose a simple yet effective way to compute the undirected adjacency matrix $E_t$ on graph $G_t$. To be specific, for items $i$ and $j$, we have edge weight $e^{t'}_{i,j}$ and $e^{t'}_{j,i}$ of 1 if user clicked item $j$ after item $i$ in any session of day $t' < t$; otherwise $e^{t'}_{i,j} = 0$. Considering all the previous days $t'$, i.e. $t' = 0, \cdots, t-1$ , element $e_{i,j}$ in matrix $E_t$ is thus defined by:

$$e_{i,j} = \frac{e^0_{i,j}}{2^{t-1}} + \frac{e^1_{i,j}}{2^{t-2}} ... + \frac{e^{t-1}_{i,j}}{2^0}. \quad (1)$$

Motivated by the memory decay in reinforcement learning, we use an exponential denominator to re-scale the edge weights. That means the historical edge weight impact will become more smaller with the increasing of time discrepancy between previous day $t'$ and current one $t$. It is notably pointed out that the defined computation on adjacency matrix $E_t$ is simple and efficient. Since we apply an exponential decay with base of 2, in the next day, $E_{t+1}$ could be simply computed by: $E_{t+1} = E(\{e^t_{i,j}\}) + E_t/2$, where $E(\{e^t_{i,j}\})$ represents the edge weight matrix newly obtained on day $t$.

Given adjacency matrix $E_t$ on graph $G_t$, for session $s$ in day $t$, we implement a multi-layer graph convolutional network (GCN) to learn the high-order item interactions. Specially, the graph convolution at the $k$-th layer is formally defined as:

$$x^k_{i,t} = \sigma\left(\frac{1}{\sum_{j\in\mathcal{N}^t_i} e_{i,j} + 1}\left(\sum_{j\in\mathcal{N}^t_i} e_{i,j}x^{k-1}_{j,t} + x^{k-1}_{i,t}\right)W_1\right). \tag{2}$$

$x^k_{i,t} \in \mathbb{R}^{1\times d}$ denotes the learned embedding of item $i$ at the $k$-th step of GCN. $\mathcal{N}^t_i$ denote the temporal neighbor set connecting to item $i$ in day $t$, where each neighbor $j$ has edge weight $e_{i,j} \geq 1$. We remove those neighbors with edge weight $e_{i,j} < 1$ to avoid the overwhelming noisy information during neighbor aggregation in GCN. $W_1 \in \mathbb{R}^{d\times d}$ is a trainable matrix in GCN. $\sigma$ is the activation function LeakyReLU. The initial embedding $x^0_{i,t}$ is given by the $i$-row of embedding layer $X$. Notably, each layer of GCN will aggregate all the first-order neighbor embedding and that of the item itself. After $K$-layer iterations, the local structural information within K-hop and the self information are contained in each item.

Based on the series of item embeddings learned from the different layers in GCN, we use an average pooling to obtain the temporal item embedding and preserve the interaction information of different orders on graph $G_t$. Formally, for item $i$ in the given session $s$, its temporal embedding is generated by:

$$x_{i,t} = \frac{1}{1+K}(x^0_{i,t} + x^1_{i,t} + \cdots, +x^K_{i,t}). \tag{3}$$

## 2.3 Session Embedding Learning Layer

This layer aims at obtaining embedding of the whole session $s$ to facilitate the following recommendation prediction task. We first process both the intra-session and temporal item embeddings learned before, and then use attention mechanism to compute session embedding.

**Normalization.** To alleviate GNN-based model popularity-biased problem [2], we apply L2 normalization on the temporal item embedding as:

$$x_{i,t} = x_{i,t}/||x_{i,t}||_2, \tag{4}$$

where $||\cdot||_2$ denote the L2 norm of an embedding vector.

**Item embedding combination.** Given the intra-session and temporal item embeddings, we simply summarize them to obtain the final item embedding, and apply dropout to avoid overfitting:

$$\hat{x}_{i,t} = \text{Dropout}(x_{i,t} + h_{i,t}). \tag{5}$$

**Session embedding.** Considering the input session $s = [i_{1,t}, i_{2,t}..., i_{l,t}]$ with length $l$, each of the final item representation is used as query to calculate attention score with the last item:

$$\beta_i = \phi(\hat{x}_{i,t}W_2 + \hat{x}_{l,t}W_3 + b)q, \quad \text{for } i = 1\cdots l. \tag{6}$$

$W_2, W_3 \in \mathbb{R}^{d\times d}$ are attention matrices; $b \in \mathbb{R}^{1\times d}$ is trainable bias; $q \in \mathbb{R}^{d\times 1}$ is a vector; and $\phi$ is sigmoid activation function. The session embedding is then given by the weighted average of its item embeddings:

$$s_t = [(\sum_{i=1}^{l} \beta_i\hat{x}_{i,t})||\hat{x}_{l,t}]W_4, \tag{7}$$

where $||$ denote the embedding concatenation, and $W_4 \in \mathbb{R}^{2d\times d}$ is trainable matrix.

## 2.4 Prediction Layer

Given session embedding $s_t$, the next clicking probabilites concerning all the candidate items are obtained by:

$$\hat{y} = \text{softmax}(s_t X^\top). \tag{8}$$

Let $y \in \mathbb{R}^{1\times I}$ denote the ground truth label vector about the next item. We treat the recommendation task as a classification problem, and train with cross-entropy loss as:

$$\mathcal{L}(y) = -\sum_{i=1}^{|I|} y_i log(\hat{y}_i) + (1 - y_i)log(1 - \hat{y}_i). \tag{9}$$

# 3 EXPERIMENTS

In this section, we conduct experiments over three benchmark datasets to answer several questions. First, how effective is TASRec compared with the state-of-the-art baselines? Second, how much does each component of TASRec contribute?

## 3.1 Experimental Setup

**Datasets.** We evaluate the proposed model on three popular session-based recommendation datasets including Diginetica[1], Retailrocket[2] and Aotm [8]. Diginetica is obtained from the CIKM Cup 2016, which records user behaviors from an E-commerce platform. Retailrocket is initially utilized for Kaggle competition, which comprises visitors' behavior data in 2015. Aotm is a music playlist dataset, which contains user's playing lists and music identifier.

Following previous experimental protocols [16] [7], we remove sessions with less than 2 interacted behaviors and delete items with less than 5 interactions for data prepossessing. Since the number of sessions between different days are unbalanced, we regard the sessions within 4, 10, and 3 successive days as one day sessions for Diginetica, Aotm, and Retailrocket datasets, respectively. In this way, we obtain 30, 36, and 46 day sessions for Diginetica, Aotm, and Retailrocket, respectively. To simulate the temporal training environment, we split data of first $\frac{5}{6}$ and last $\frac{1}{6}$ days into training and testing sets. For the training set, we use the last 20% remaining sessions for validation similar to [1]. Specifically, given a session record $s = \{i_1, i_2, \ldots, i_l\}^l_{t=2}$, we generate a set of training samples $\{(i_1, \ldots, i_{t-1}), i_t\}$, and the goal is to accurately predict the $t^{\text{th}}$ item based on the previous ones.

**Competitors.** To validate the effectiveness of TASRec, we include two categories of baselines as follows. First, to validate the usefulness of capturing interest shifting within sessions, we include three

[1]http://www.cikmconference.org
[2]https://www.kaggle.com/retailrocket/ecommerce-dataset

**Table 1: Comparisons with other baseline methods**

| Method | Aotm | | | Diginetica | | | Retailrocket | | |
|---|---|---|---|---|---|---|---|---|---|
| | Recall@20 | Recall@50 | NDCG@100 | Recall@20 | Recall@50 | NDCG@100 | Recall@10 | Recall@20 | NDCG@100 |
| GRU4REC | 1.70 | 3.64 | 1.32 | 28.90 | 40.02 | 15.66 | 17.41 | 23.11 | 12.75 |
| SR-GNN | 3.82 | 6.94 | 2.73 | 48.74 | 65.37 | 28.41 | 36.34 | 45.58 | 28.93 |
| CSRM | 3.97 | 7.30 | 2.86 | 48.88 | 66.90 | 28.34 | 37.43 | 47.56 | 28.43 |
| LESSR | 3.11 | 5.58 | 2.31 | 49.59 | 66.12 | 29.03 | 37.97 | 46.59 | 29.84 |
| TASRec | **4.42** | **7.71** | **3.08** | **50.39** | **67.27** | **29.48** | **39.08** | **48.16** | **30.36** |
| Improv. | **11.34%** | **5.62%** | **7.69%** | **1.61%** | **0.55%** | **1.55%** | **2.92%** | **1.26%** | **1.74%** |

**Table 2: Ablation study results on Retailrocket dataset**

| Model | Recall@10 | Recall@20 | Recall@50 | NDCG@100 |
|---|---|---|---|---|
| TASRec-temp | 36.73 | 46.65 | 58.40 | 27.55 |
| TASRec-base | 36.60 | 45.45 | 56.49 | 28.84 |
| TASRec | **39.08** | **48.16** | **59.48** | **30.36** |

standard static session models, GRU4REC [3],SR-GNN [17], and LESSR [1]. Second, to demonstrate the superiority of the proposed method over other temporal session model, we include a recent proposed model CSRM [15].

For our model, we implement it with Pytorch and the Adam optimizer is used to optimize the model, where the initial learning rate is set to 0.001 and will decay after two epochs. To avoid over-fitting problem, dropout [11] and weight regularization [4] techniques are also adopted. For all methods, we applied grid search method following [1] [12] to tune the hyper-parameters using validation set. Specifically, we search embedding dimension $d$ from $\{50, 100, 150, 200\}$ while the number of graph neural network layers $K$ from $1, 2, 3, 4, 5$. And the batch size is set to 128 for all models. The best results are reported in our experiments.

**Evaluation Metrics.** To evaluate the performance of all methods, we adopt two widely used evaluation metrics, i.e., Recall@$N$ and NDCG@$N$ (Normalized Discounted Cumulative Gain). In particular, Recall measures the predictive accuracy of top-$N$ ranking list, while NDCG estimates the ranking order of the list.

## 3.2 Experimental Results

We first start to answer the first question proposed at the beginning of this section. Table 1 summarizes the results on three datasets in terms of Recall and NDCG values.

From the results, we have several observations. First, TASRec performs consistently better than other baselines across three datasets. It validates the effectiveness of the proposed method. Second, CSRM achieves generally better or comparable performance than three static methods (GRU4REC, SR-GNN, and LESSR), which demonstrates the usefulness to capture the interest shifting in session-based recommendation scenarios. However, CSRM is outperformed by our model in all cases. The primary reason is that our model explicitly utilizes a temporal modeling layer to capture user's interest shifting while CSRM relies on the query quality to retrieve relevant items. This comparison indicates the effectiveness of our model in modeling user's dynamic interests over time. Third, the improvement between our model and the second best method decreases when the number of recalled items increases. It indicates that our model is capable of ranking the most likely clicked items in the top ranking list. These results validate the effectiveness of the proposed method.

## 3.3 Ablation study

We now start to investigate the second question, i.e., how much does each component of TASRec contribute. To this end, we introduce two variants, i.e., TASRec-temp and TASRec-base. TASRec-temp is obtained by excluding the intra-session modeling layer, which is used to validate the effectiveness of capturing intra-session sequential structures. TASRec-base is obtained by removing the temporal modeling layer, which is derived to validate the usefulness of capturing user's interest shifts. The results on Retailrocket dataset are summarized in Table 2.

From the results in Table 2, we observe that TASRec achieves significantly better performance than its two variants. Specifically, the improvement between TASRec and TASRec-temp validates the effectiveness of modeling the sequential patterns within the session for session-based recommendations. The gap between TASRec and TASRec-base demonstrates the effectiveness of the proposed temporal modeling layer in capturing the dynamics of user interests over time.

# 4 CONCLUSIONS

In this paper, we propose a novel memory-efficient framework, dubbed TASRec, for session-based recommendations when the number of available sessions is large. TASRec constructs a dynamic graph to preserve the item-to-item relationships over time, and then builds a tailored graph neural network to learn the temporal augmented item representations, which are fed into the standard sequential neural networks for effective recommendation. TASRec not only can capture the dynamics of user interests but also is memory efficient during the training. Empirical results on three real-world datasets demonstrate the effectiveness of TASRec compared with state-of-of-the art baselines. Our future work is to explore user's diverse interests [6, 14] for more accurate recommendation.

# 5 ACKNOWLEDGMENTS

This work is jointly supported by PolyU (UGC) Start-up Fund (#P0033934) and NSFC grants (No. 61972155), Zhejiang Lab under No.2019KB0AB04.

# REFERENCES


- [1] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling Information Loss of Graph Neural Networks for Session-based Recommendation. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash (Eds.). ACM, 1172–1180. https://dl.acm.org/doi/10.1145/3394486.3403170
- [2] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam Shroff. 2019. NISER: Normalized Item and Session Representations with Graph Neural Networks. *CoRR* abs/1909.04276 (2019). arXiv:1909.04276 http://arxiv.org/abs/1909.04276
- [3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1511.06939
- [4] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. 2017. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686* (2017).
- [5] 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, CIKM 2017, Singapore, November 06 - 10, 2017*, Ee-Peng Lim, Marianne Winslett, Mark Sanderson, Ada Wai-Chee Fu, Jimeng Sun, J. Shane Culpepper, Eric Lo, Joyce C. Ho, Debora Donato, Rakesh Agrawal, Yu Zheng, Carlos Castillo, Aixin Sun, Vincent S. Tseng, and Chenliang Li (Eds.). ACM, 1419–1428. https://doi.org/10.1145/3132847.3132926
- [6] Ninghao Liu, Qiaoyu Tan, Yuening Li, Hongxia Yang, Jingren Zhou, and Xia Hu. 2019. Is a single vector enough? exploring node polysemy for network embedding. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 932–940.
- [7] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. 2019. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys 2019, Copenhagen, Denmark, September 16-20, 2019*, Toine Bogers, Alan Said, Peter Brusilovsky, and Domonkos Tikk (Eds.). ACM, 462–466. https://doi.org/10.1145/3298689.3347041
- [8] Brian McFee and Gert R. G. Lanckriet. 2012. Hypergraph Models of Playlist Dialects. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, Fabien Gouyon, Perfecto Herrera, Luis Gustavo Martins, and Meinard Müller (Eds.). FEUP Edições, 343–348. http://ismir2012.ismir.net/event/papers/343-ismir-2012.pdf
- [9] Fei Mi and Boi Faltings. 2020. Memory Augmented Neural Model for Incremental Session-based Recommendation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, Christian Bessiere (Ed.). ijcai.org, 2169–2176. https://doi.org/10.24963/ijcai.2020/300
- [10] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. 2020. GAG: Global Attributed Graph Neural Network for Streaming Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 669–678. https://doi.org/10.1145/3397271.3401109
- [11] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR* 15, 1 (2014), 1929–1958.
- [12] Qiaoyu Tan, Ninghao Liu, Xing Zhao, Hongxia Yang, Jingren Zhou, and Xia Hu. 2020. Learning to Hash with Graph Neural Networks for Recommender Systems. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen (Eds.). ACM / IW3C2, 1988–1998. https://doi.org/10.1145/3366423.3380266
- [13] Qiaoyu Tan, Jianwei Zhang, Ninghao Liu, Xiao Huang, Hongxia Yang, Jignren Zhou, and Xia Hu. 2021. Dynamic memory based attention network for sequential recommendation. *arXiv preprint arXiv:2102.09269* (2021).
- [14] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. 2021. Sparse-Interest Network for Sequential Recommendation. In *WSDM '21, The Fourteenth ACM International Conference on Web Search and Data Mining, Virtual Event, Israel, March 8-12, 2021*, Liane Lewin-Eytan, David Carmel, Elad Yom-Tov, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 598–606. https://doi.org/10.1145/3437963.3441811
- [15] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 345–354. https://doi.org/10.1145/3331184.3331210
- [16] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xianling Mao, and Minghui Qiu. 2020. Global Context Enhanced Graph Neural Networks for Session-based Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 169–178. https://doi.org/10.1145/3397271.3401142
- [17] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-Based Recommendation with Graph Neural Networks. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 346–353. https://doi.org/10.1609/aaai.v33i01.3301346
- [18] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. 2019. Graph Contextualized Self-Attention Network for Session-based Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 3940–3946. https://doi.org/10.24963/ijcai.2019/547