Exploiting Explicit and Implicit Item relationships for Session-based Recommendation

Zihao Li*
University of Technology Sydney
Sydney, Australia
zihao.li@student.uts.edu.au

Xianzhi Wang
University of Technology Sydney
Sydney, Australia
xianzhi.wang@uts.edu.au

Chao Yang
University of Technology Sydney
Sydney, Australia
chao.yang@student.uts.edu.au

Lina Yao
University of New South Wales
Sydney, Australia
lina.yao@unsw.edu.au

Julian McAuley
University of California, San Diego
La Jolla, CA, USA
jmcauley@ucsd.edu

Guandong Xu
University of Technology Sydney
Sydney, Australia
guandong.xu@uts.edu.au

ABSTRACT

The session-based recommendation aims to predict users' immediate next actions based on their short-term behaviors reflected by past and ongoing sessions. Graph neural networks (GNNs) recently dominated the related studies, yet their performance heavily relies on graph structures, which are often predefined, task-specific, and designed heuristically. Furthermore, existing graph-based methods either neglect implicit correlations among items or consider explicit and implicit relationships altogether in the same graphs. We propose to decouple explicit and implicit relationships among items . As such, we can capture the prior knowledge encapsulated in explicit dependencies and learned implicit correlations among items simultaneously in a flexible and more interpretable manner for effective recommendations. We design a dual graph neural network that leverages the feature representations extracted by two GNNs: a graph neural network with a single gate (SG-GNN) and an adaptive graph neural network (A-GNN). The former models explicit dependencies among items . The latter employs a self-learning strategy to capture implicit correlations among items. Our experiments on four real-world datasets show our model outperforms state-of-the-art methods by a large margin, achieving 18.46% and 70.72% improvement in HR@20, and 49.10% and 115.29% improvement in MRR@20 on *Diginetica* and *LastFM* datasets.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

session-based recommendation, graph neural network, explicit and implicit relationships

*Zihao Li is the corresponding author.

WSDM '23, February 27-March 3, 2023, Singapore, Singapore

ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570432

ACM Reference Format:
Zihao Li, Xianzhi Wang, Chao Yang, Lina Yao, Julian McAuley, and Guandong Xu. 2023. Exploiting Explicit and Implicit Item relationships for Session-based Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27-March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570432

1 INTRODUCTION

Session-based recommendation focuses on capturing users' short-term interests from sessions rather than exploring their rich historical interactions or modeling users' long-term interests [29]. It has shown significant advantages in dynamic and real-time recommendations. Existing session-based recommendation methods are mainly sequence-based or graph-based. Sequence-based methods view items in a session as a sequence and predict the next item with which users may interact. For example, Markov Chain-based methods [21, 23] map the current session into a Markov Chain and then infer the user's next action solely based on the last item in the session. Deep learning models like RNN, GRU and LSTM [9, 10, 19, 25] are increasingly applied to session-based recommendation, given their outstanding feature representation ability. Several studies, e.g., NARM [12], SHAN [37], STAMP [15], and Transformers4rec [6], further apply attention mechanisms to distinguish the importance of items and capture user intentions in session-based recommendation. Graph-based methods rely on graph structures to represent relationships among items and aggregate the auxiliary information to improve performance. The graphs can be *intra-session* or *inter-session*. The former only considers the item relations within a single session [3, 7, 16] while the latter considers more than one session in the same recommendation problem [32, 41].

Sequential models generally become ineffective when dealing with short sessions, which widely exist [2, 18]. For example, the average and median lengths of sessions in a popular session dataset, *Diginetica*[1] are only 4.80 and 4.00. The lengths are 3.97 and 3.00 for another dataset, *Yoochoose*[2]. Graph-based methods rely on problem-specific designs of graph structures to achieve good performance. Although predefined graphs contain prior domain knowledge, they may be incomplete or even unavailable due to the difficulty of capturing latent and long-range dependence among sessions [38]. Furthermore, existing graph-based methods focus on modeling

[1]http://cikm2016.cs.iupui.edu/cikm-cup
[2]http://2015.recsyschallenge.com/challege.html

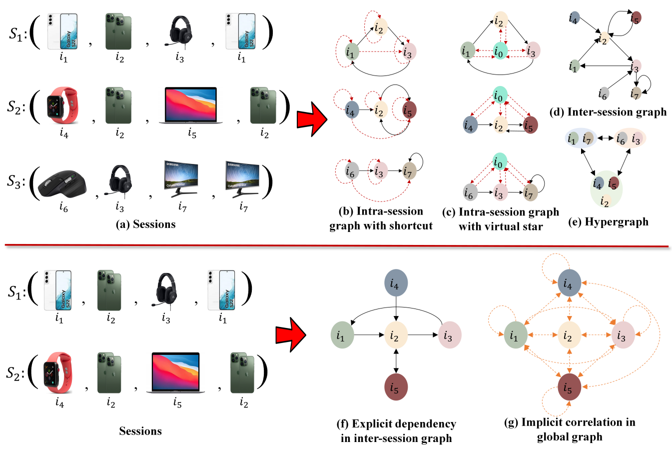


Figure 1: The upper half shows different graph structures for modeling item relationships in an example of three sessions. (b) applies shortcuts and self-loops (denoted by red dotted lines) to each session to capture long-range dependencies [3]; (c) creates a virtual item (i_0) to connect all the items in each session [16]; (d) illustrates all item relationships across all sessions with a single graph [36]; (e) groups items (e.g., according to their brands) and builds a hypergraph based on items' co-occurrence in the same sessions [34]. The lower half showcases our proposal of decoupling explicit and implicit item relationships for an example of two sessions.

explicit dependencies while neglecting the implicit relationships, which are proven equally important [5, 33].

We aim to leverage the explicit dependencies and dynamic correlations among items as reflected by sessions simultaneously for effective session-based recommendation. To this end, we propose a novel **D**ual **G**raph **N**eural **N**etwork (DGNN), which models explicit dependencies and implicit correlations between items separately for node representation and next item prediction. Decoupling explicit dependencies and implicit correlations could provide extra flexibility for model training [35] and improves interpretability through visualization. Here, we use examples (Figure 1, detailed in Section 3) to illustrate the distinctions of our approach from previous approaches in graph structure.

In a nutshell, we make the following contributions in this paper:

- To the best of our knowledge, we are the first to decouple explicit and implicit relationships among items in a holistic approach for effective session-based recommendation.
- We propose an adaptive graph neural network (A-GNN) to capture the implicit correlations between items in a self-learning strategy. This allows the graph structures to dynamically change during model training to accommodate the evolution of users' preferences.
- We present a novel graph neural network with a single gate (SG-GNN) to harness the explicit ordering (or sequential dependencies) of items in an inter-session graph.
- Our extensive experiments on four real-world datasets demonstrate our model outperforms several baselines and state-of-the-art methods. All datasets and code will be made public via GitHub[3].

2 RELATED WORK

2.1 Sequential Models

Sequential Neural Network-based Methods. Several studies leverage sequential information explicitly via sequential neural networks, e.g., LSTM, and GRU, to capture users' historical interests [10, 19, 25, 28]. For instance, GRU4Rec [10] is the first that applied RNNs to model the whole session for the next item recommendation. HRNN [19] develops hierarchical RNNs with inter-session information transfer to address the problem of personalizing session-based recommendation. NARM [12], a neural attentive recommendation machine, applies a hybrid encoder with an attention mechanism to model users' sequential behavior and capture the main purpose in the current session. Liu et al. [15] follow a similar idea but replace the recurrent neural network with a simple multilayer perceptron (MLP) network, and propose a short-term attention/memory priority (STAMP) model to capture both long-term and short-term interests.

2.2 Graph Models

Intra- and Inter- session Graph Methods. Graph neural networks can capture multi-hop contextual information and learn item correlations and representation through information propagation and aggregation. Due to the above merits, GNNs achieve promising results on session-based recommendation. Intra-session graph methods only rely on the current session for graph construction. Wu et al. [33] propose SR-GNN, which is a pioneering work that introduces a graph neural network to model explicit relationships between items for session-based recommendation. Based on this work, Xu *et al.* [36] adopt GNN to extract local context information and apply a self-attention network to capture global dependencies between distant items. Although intra-session-based models can explore rich transitions between items, they only leverage the current session to make the recommendation.

Compared with intra-session graphs, an inter-session graph utilizes the current and neighbor sessions for recommendation. Specifically, inter-session graph methods can be divided into user-oriented models [4, 39], item-oriented models [1, 11, 32, 36, 41], and others [7, 17, 34]. User-oriented models create the inter-session graph via users' historical interaction records or social information. For instance, Chen et al. [4] consider users' friends and the social influences for recommendation. As for the item-oriented models, most related work constructs the inter-session graph via similar attributes of items or adjacent relationships between items across all historical sessions. Wang et al. [32] and Huang et al. [11] exploit all item transitions to better infer user preferences in the current session. Furthermore, some other works also explore hypergraph [7, 13, 34] or heterogeneous graph [8, 17] for session-based recommendation.

[3]https://github.com/ZihaoLi97/WSDM23-DGNNs--for-Session-based-Recommendation

GNN with Implicit Relationships Modeling. Convolutional GNNs (e.g., SR-GNN and GC-SAN) only aggregate information from adjacent items while neglecting the information from all other items. Pan et al. [16] introduce a virtual star node in the current session to capture non-adjacent items' information and achieve long-range information propagation. Chen et al. [3] propose two graphs with self-loop and shortcuts for items' co-occurrence and order modeling to alleviate the information loss and long-range dependency capturing problems. Furthermore, based on the sub-cluster of sessions, Guo et al. [7] construct multi-granularity intent units for each session to capture users' multi-level interests through a fixed graph structure. However, as users' multi-granularity intent [7, 24], the false adjacent [40] and long-range dependency [3] in graphs, fixed graph structures are insufficient and inflexible to adapt to all the datasets, which will incur the model to achieve sub-optimal results. The above motivates us to propose an adaptive graph convolution module to model the implicit correlations with a self-learning strategy for session-based recommendation.

3 PROBLEM FORMULATION

Let $\mathcal{I} = \{i_1, i_2, i_3, ...i_N\}$ be the set of items, where N is the number of items. Each session $s = [i_1, i_2, i_3, ..., i_o]$ consists of a sequence of interactions $i_k \in \mathcal{I}(1 \le k \le o)$ related to one user. Suppose we embed each item $i \in \mathcal{I}$ into the same space and denote by $\mathbf{x}_i \in \mathbb{R}^d$ the representation of item i. Therefore, the representation of the item set is denoted by $\mathbf{X} \in \mathbb{R}^{N \times d}$.

Given a session s, session-based recommendation aims to predict the next click item $i_{s,m+1}$. Our model generates probabilities $\hat{\mathbf{y}}$ for all possible items based on the input session s. Each element's value of vector $\hat{\mathbf{y}}$ is the recommendation score of the corresponding item. The items with the top-k recommendation scores will be recommended as the model's output.

Graph for explicit dependencies. As shown in Figure 1(f), given a session set $\mathcal{S}$, we denote the relationships between all adjacent items in $\mathcal{S}$ via an inter-session graph (dynamic global graph) $\mathcal{G}^s = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ indicate the set of nodes and edges respectively. Each node represents an item in sessions. For an edge $e \in \mathcal{E}$, it could be represented by an ordered tuple (v_i, v_j) (v_i, v_j are adjacent items in sessions) which indicates the edge points from node v_i to node v_j. Hence, we define an explicit dependency from v_i to v_j. The connectivity among the whole graph is represented by an adjacency matrix $\mathbf{A}^s \in \mathbb{R}^{N \times N}$ with $\mathbf{A}^s_{ij} \neq 0$ iff $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}^s_{ij} = 0$ iff $(v_i, v_j) \notin \mathcal{E}$, where N is the total number of nodes. In addition, the adjacent matrix A^s is normalized following $\tilde{\mathbf{A}}^s = \mathbf{A_{ij}}^s / \sum_j \mathbf{A_{ij}}^s$. Hence, the inter-session graph could model the explicit dependency between items.

Graph for implicit correlations. As shown in Figure 1(g), given a session set $\mathcal{S}$, we could also construct a global graph $\mathcal{G}^g = (\mathcal{V}, \mathcal{E})$. We add an edge between any pair of nodes in $\mathcal{G}^g$ to indicate an implicit correlation. The element A^g_{ij} in an adjacency matrix $\mathbf{A}^g$ represents the correlation between item i and item j, which can be learned and adjusted dynamically by A-GNN. To decrease the time and space complexity, in our paper, we construct a global graph for all the sessions in a batch, i.e., the sessions from the same batch will share a same global graph.

4 METHODOLOGY

The overall architecture of our proposed approach (Figure 2) consists of four major components: an adaptive graph neural network for implicit information aggregation and node representation (Section 4.2), a graph neural network with a single gate for explicit information aggregation and node representation (Section 4.3), a session representation layer (Section 4.4), a prediction layer and the loss function (Section 4.5).

4.1 Overview

Our framework works as follows. First, it constructs an inter-session graph. Specifically, we collect an item set based on neighbor sessions from one batch, and convert every item $v_i \in \mathcal{V}$ into a unified low-dimension embedding space $\mathbf{X}$. Then, the item embedding is fed into a dual graph neural network (A-GNN and SG-GNN) to capture implicit and explicit item relationships. The fusion layer fuses the updated item representation $\tilde{\mathbf{X}}$ from those two modules. Finally, our soft-attention mechanism obtains session representations $\mathbf{s}$ and a *softmax* function generates the next item's predication probability $\hat{\mathbf{y}}$. We formulate the above process as follows:

$$
\begin{aligned}
&\mathbf{X}^{(m)}_{\text{A-GNN}} = \text{A-GNN}(\mathbf{X} + \mathbf{X}^{(1)}_{\text{A-GNN}} + ... + \mathbf{X}^{(m-1)}_{\text{A-GNN}}, \mathbf{A}^g)\\
&\mathbf{X}^{(l)}_{\text{SG-GNN}} = \text{SG-GNN}(\mathbf{X}^{(l-1)}_{\text{SG-GNN}}, \tilde{\mathbf{A}}^s)\\
&\tilde{\mathbf{X}} = \text{F}(\mathbf{X}^{(m)}_{\text{A-GNN}}, \mathbf{X}^{(l)}_{\text{SG-GNN}})\\
&\mathbf{s} = \text{SR}(\tilde{\mathbf{X}})\\
&\hat{\mathbf{y}} = \text{P}(\mathbf{s}, \mathbf{X})
\end{aligned}
\tag{1}
$$

where $\mathbf{X}^{(m)}_{\text{A-GNN}}$ is the item representation of A-GNN with m blocks. $\mathbf{X}^{(l)}_{\text{SG-GNN}}$ is the item representation of SG-GNN after l convolution layers. $\text{F}(\cdot), \text{SR}(\cdot), \text{P}(\cdot)$ are the representation fusion layer, session representation layer and prediction layer, respectively.

4.2 A-GNN Module

The adaptive graph neural network (A-GNN) module aims to capture implicit correlations between any of two items dynamically with a self-learning strategy for item representation. To achieve this goal, A-GNN employs multi-head correlation, formulated below,

$$
\begin{aligned}
&\mathbf{Q}_i = \mathbf{X}\mathbf{W}^Q_i, \quad \mathbf{K}_i = \mathbf{X}\mathbf{W}^K_i, \quad \mathbf{V}_i = \mathbf{X}\mathbf{W}^V_i\\
&\mathbf{A}^g_i = \text{Dropout}(\tanh(\mathbf{Q}_i\mathbf{K}^T_i))\\
&\mathbf{X}_{\text{A-GNN}_i} = \mathbf{A}^g_i\mathbf{V}_i\\
&\mathbf{X}_{\text{A-GNN}} = \text{Dropout}(\text{ReLU}([\mathbf{X}_{\text{A-GNN}_0}||...||\mathbf{X}_{\text{A-GNN}_k}])\mathbf{W}^M)
\end{aligned}
\tag{2}
$$

where $\mathbf{X} \in \mathbb{R}^{N \times d}$ is the item embedding, $\mathbf{X}_{\text{A-GNN}}$ is the item representation generated by A-GNN, and $\mathbf{A}^s$ is the adjacency matrix of the dynamic global graph. Each element of $\mathbf{A}^g$, say $\mathbf{A}^g_{ij}$, represents the correlation between item i and item j. $\mathbf{W}^Q_i, \mathbf{W}^k_i, \mathbf{W}^V_i, \mathbf{W}^M$ are all learnable parameter matrices. || is a concatenation operation, and k is the head number (in this paper, we set $k = 4$). Compared with existing self-attention mechanisms [26], A-GNN replaces *softmax* with a *tanh* function to cope with non-positive correlations between items. It also differs from GAN [27] in obtaining the correlations between any pair of items rather than with the neighbor items.

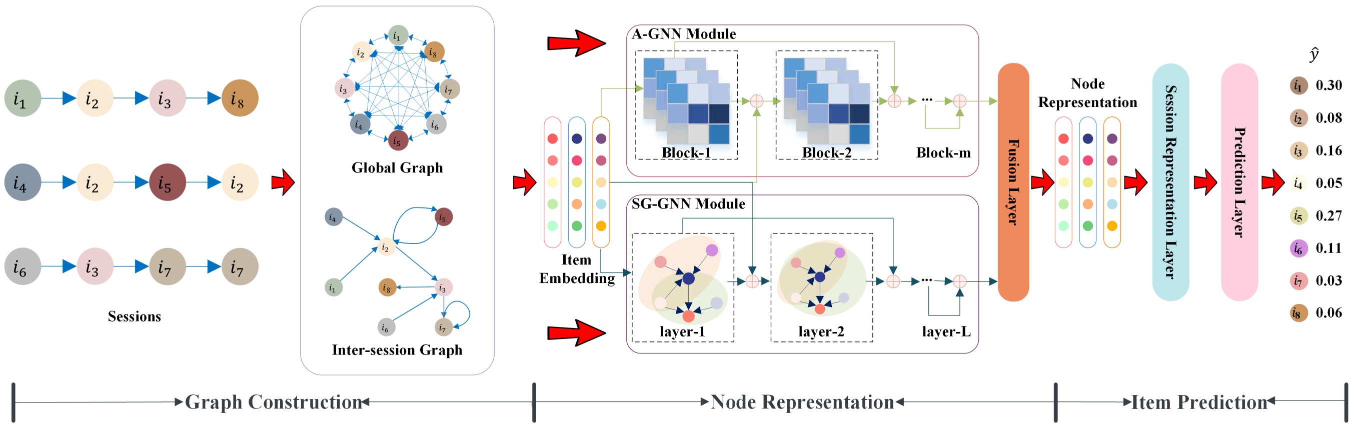


Figure 2: Architecture of DGNN.

We stack multiple A-GNN blocks to enhance the model's representation capacity. As such, each module takes all the previous blocks' outputs as the input:

$$\mathbf{X}^{(m)}_{\text{A-GNN}} = \text{A-GNN}(\mathbf{X} + \mathbf{X}^{(1)}_{\text{A-GNN}} + \ldots + \mathbf{X}^{(m-1)}_{\text{A-GNN}}, \mathbf{A}^{g}) \tag{3}$$

where A-GNN$(\cdot)$ denotes the A-GNN block, m is the number of A-GNN blocks, and $\mathbf{X}^{(i)}_{\text{A-GNN}}$ is the i-th block's output. A-GNN's final output is the representation of the last block M, i.e., $\mathbf{X}^{(M)}_{\text{A-GNN}}$.

4.3 SG-GNN Module and Fusion Layer

The graph neural network with a single gate (SG-GNN) module aims to leverage the explicit dependencies among items as reflected by the sequential information in sessions. To this end, SG-GNN aggregates the information of neighbors into the center node via a gate mechanism for representation update:

$$\begin{aligned} &\mathbf{J} = \mathbf{X}\mathbf{W}_J, \quad \mathbf{P} = \mathbf{X}\mathbf{W}_P, \quad \mathbf{Z} = \mathbf{X}\mathbf{W}_Z \\ &\mathbf{R} = \tilde{\mathbf{A}}^{s}\mathbf{J}\mathbf{W}_R, \quad \mathbf{U} = \tilde{\mathbf{A}}^{s}\mathbf{J}\mathbf{W}_U \\ &\mathbf{X}_{\text{SG-GNN}} = \mathbf{P} + \text{ReLU}(\mathbf{R} + \mathbf{Z}) \odot \mathbf{U} \end{aligned} \tag{4}$$

where $\tilde{\mathbf{A}}^{s}$ is the normalized adjacency matrix (defined in Section 3). $\mathbf{W}_J$, $\mathbf{W}_R$, $\mathbf{W}_Z$, $\mathbf{W}_P$, $\mathbf{W}_U$ are learnable parameter matrices. The gate mechanism controls how much information from neighbors is considered for updating node representation.

We apply multi-layers graph convolution as described below. In particular, we use item embedding $\mathbf{X}$ as the first layer's input. SG-GNN's final output is the last layer's item representation $\mathbf{X}^{(L)}_{\text{SG-GNN}}$.

$$\mathbf{X}^{(l)}_{\text{SG-GNN}} = \text{SG-GNN}(\mathbf{X}^{(l-1)}_{\text{SG-GNN}}, \tilde{\mathbf{A}}^{s}) \tag{5}$$

Given implicit and explicit representations of items from A-GNN and SG-GNN, we fuse them using a linear projection, thus obtaining the final item representation $\tilde{\mathbf{X}}$ as follows.

$$\tilde{\mathbf{X}} = [\mathbf{X}^{(M)}_{\text{A-GNN}} || \mathbf{X}^{(L)}_{\text{SC-GNN}}]\mathbf{W}_F \tag{6}$$

where $\mathbf{W}_F \in \mathbb{R}^{2d \times d}$ is a learnable parameter matrix.

4.4 Session Representation Layer

We use local and global representations of sessions to capture users' short-term and long-term preferences. Given a session $s = [i_1, i_2, \ldots, i_m]$, we assume users' current preference can be reflected by the last item i_m, following previous research [33]. We thereby use the representation of the last-clicked item i_m as the session's local representation, i.e., $\mathbf{s}_l = \tilde{\mathbf{x}}_m$. As for the global representation of session s, $\mathbf{s}_g$, we generate it based on the representations of all items in the session. Specifically, we employ a soft-attention mechanism to fuse the information from all items while taking into account their varied importance:

$$\begin{aligned} \alpha_i &= \mathbf{q}^T \sigma(\mathbf{W}_1 \tilde{\mathbf{x}}_m + \mathbf{W}_2 \tilde{\mathbf{x}}_i + \mathbf{c}) \\ \mathbf{s}_g &= \sum_{i=1}^{m} \alpha_i \tilde{\mathbf{x}}_i \end{aligned} \tag{7}$$

where $\mathbf{q}^T, \tilde{\mathbf{x}}_i \in \mathbb{R}^d$ and $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ are all learnable parameters. α controls the weights of item representations.

Finally, we concatenate the local and global representations via a linear transformation to obtain the final session representation $\mathbf{s}$:

$$\mathbf{s} = [\mathbf{s}_l || \mathbf{s}_g]\mathbf{W}_3 \tag{8}$$

where matrix $\mathbf{W}_3 \in \mathbb{R}^{2d \times d}$ compresses two combined representation vectors into the latent space $\mathbb{R}^d$.

4.5 Prediction Layer and Loss Function

We calculate item scores $\hat{\mathbf{z}} \in \mathbb{R}^N$ as the inner production of item embedding $\mathbf{X}$ and session representation $\mathbf{s}$:

$$\hat{\mathbf{z}} = \mathbf{s}^T \mathbf{X} \tag{9}$$

Then, we apply a *softmax* function to the scores for next-item prediction. This will generate probabilities indicating how likely each item would be the next to be clicked by the user:

$$\hat{\mathbf{y}} = \text{softmax}(\hat{\mathbf{z}}) \tag{10}$$

For each session, we define the loss function as the cross-entropy of the prediction and the ground truth:

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{n} \mathbf{y}_i log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) log(1 - \hat{\mathbf{y}}_i) \tag{11}$$

where $\mathbf{y}$ is the one-hot encoding of the ground-truth item.

5 EXPERIMENTS

In this section, we report our experimental setting, including datasets, baselines, evaluation metrics, and an analysis of experimental results. We aim to answer the following questions:

- **RQ1.** How does the DGNN perform compared with state-of-the-art (SOTA) session-based recommendation methods?
- **RQ2.** How do different sub-modules in the DGNN affect recommendation performance?
- **RQ3.** How do hyper-parameter settings influence model performance?
- **RQ4.** How is the model interpretation capability of DGNN?

5.1 Datasets

We conducted experiments on four real-work datasets commonly used for session-based recommendation.

- *Diginetica* is a personalized e-commerce research challenge dataset from CIKM CUP 2016. The dataset contains transition history, which is suitable for session-based recommendation. Following [1, 3, 15, 20, 33], we used the sessions in the last week for test.
- *Yoochoose* is a dataset that contains a stream of user clicks on an e-commerce website within six months from RecSys Challenge 2015. We conducted the typical method in [1, 3, 15, 20, 31, 33] to split the dataset. Since the training set of Yoochoose is extremely large, we used the most recent portions 1/64 and 1/4 subsample of all training sessions as the training set, denoted as "Yoochoose1/64" and "Yoochoose1/4", respectively.
- *Gowalla* is a popular dataset widely used for point-of-interest recommendation. Following [1, 3, 7], we kept the top 30,000 most popular locations and grouped users' check-in records into disjoint sessions by splitting intervals between adjacent records that are longer than one day. We used the last 20% of sessions as the test set.
- *Last.FM* is a music artist recommendation dataset. Following [1, 3, 7, 20], we kept the top 40,000 most popular artists and treated users' transactions in 8 hours as a session. Like Gowalla, we used the most recent 20% of sessions as the test set.

Following [1, 3, 7, 20, 33], we filtered out sessions of length 1 and items appearing less than 5 times. Furthermore, for each given session $s = [i_1, i_2, ..., i_m]$, we generated the input and corresponding labels, i.e., $([i_1], i_2), ([i_1, i_2], i_3), ..., ([i_1, i_2, ..., i_{m-1}], i_m)$, for all the datasets. Table 1 summarizes the statistics for the datasets.

Table 1: Statistics of datasets

	Diginetica	Yoochoose1/64	Yoochoose1/4	Gowalla	Last.FM
#clicks	981,620	557,248	8,326,407	1,122,788	3,835,706
#train sessions	716,835	369,859	5,917,745	675,561	2,837,644
#test sessions	60,194	55,898	55,898	155,332	672,519
#items	42,596	16,766	29,618	29,510	38,615
#length $\leq$ 5	537,546	289,490	4,234,915	627,100	1,136,909
#length >5	239,483	136,267	1,738,734	203,793	2,373,254
Average length	4.80	6.16	5.71	4.32	9.16

5.2 Baselines and Evaluation Metrics

We chose 11 baselines from five categories of methods: conventional methods (e.g., popularity- and Nearest Neighbors (NN)-based methods), sequence-based methods including Markov Chain and sequential neural networks, graph neural networks with intra-session graphs and inter-session graphs, as listed below:

- **POP** is a simple benchmark that recommends the most popular (highest ranked) item for users.
- **Item-KNN** [22] recommends items through the similarity between every item of the current session and the other items.
- **FPMC** [4] [21] combines the first-order Markov Chain with matrix factorization to capture both sequential effects and user preferences.
- **GRU4Rec**[5] [10] employs a gated recurrent unit to model the sequential behavior of items in a session.
- **NARM**[6] [12] improves GRU4Rec by introducing RNN with attention to session-based recommendation.
- **SR-GNN** [7] [33] models explicit dependencies within a session via a graph neural network and then applies a soft-attention mechanism to generate session-level embeddings.
- **SGNN-HN** [16] applies a star graph neural network to model the complex transition relationships between items without direct connections in an ongoing session.
- **LESSR** [8] [3] introduces two kinds of session graphs with self-loop and short-cuts to capture implicit connections and solve the information loss and long-rage dependency problem.
- **MSGIFSR** [9] [7] proposes a consecutive intent unit to extract user intent from different granularities based on different item groups in the current session. It achieves the latest SOTA in the above four datasets.
- **GC-SAN** [36] gets local context information by using GGNN and then utilizes a self-attention mechanism to capture explicit dependency.
- **GCE-GNN** [32] consider ϵ-neighbor ($\epsilon = 2$) connections to construct an inter-session graph for session-based recommendation.

We evaluated all models with two widely used metrics: HR@20 (Hit Rate) and MRR@20 (Mean Reciprocal Rank). HR@20 represents the proportion of correctly recommended items among the top 20 items. MRR@20 is the average of reciprocal ranks of the correctly-recommended items. The reciprocal rank is set to 0 when the rank exceeds 20.

[4] https://github.com/khesui/FPMC
[5] https://github.com/hidasib/GRU4Rec
[6] https://github.com/lijingsdu/sessionRec_NARM
[7] https://github.com/CRIPAC-DIG/SR-GNN
[8] https://github.com/twchen/lessr
[9] https://github.com/SpaceLearner/SessionRec-pytorch

5.3 Experimental Setup

For a fair comparison, we followed [7, 16, 33] and selected the Adam optimizer with the initial learning rate of 0.001, which will decay by 0.5 after every five epochs. We set the L_2 regularization to 10^{-5} and used an early stopping strategy (no improvements in the evaluation metrics for five consecutive epochs) to relieve the overfitting problem. We initialized all parameters using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. We fixed both the embedding dimension and batch size at 100. For the GC-GNN module, the number of layers varied within {1,2,3,4}. For the A-GNN module, the block number was within the scope of {4,5,6}. We tested the dropout ratio within {0.1, 0.5, 0.9}.

5.4 Overall Comparison (RQ1)

Our comparison results (Table 2) show our method (DGNN) significantly outperformed all the baselines, which is largely attributed to the two modules' capability to capture more accurate and complete user preferences—while SG-GNN can effectively integrate the explicit information from neighbors through the improved graph convolution operation, the self-learning dynamic graph can learn implicit correlations between items, which are equally important for improving the recommendation performance. In particular, DGNN outperformed state-of-the-art (SOTA) performance by a large margin, i.e., a 115.29% improvement, on *Last.FM*, which contains longer sessions when compared with other datasets. This reveals the ability of graph neural networks to handle prediction tasks on long-range sessions when equipped with explicit and implicit item relationship modeling.

Deep learning methods performed significantly better than traditional methods (e.g., POP, Item-KNN, and FPMC), demonstrating their superior complex feature extraction and representation ability. NARM outperformed GRU4Rec because NARM can not only capture the latent sequential information in sessions (as GRU4Rec does) but also learn item correlations via the attention mechanism. GNN-based models generally outperformed sequence-based methods, showing the importance of session graphs in representing transition relationships between different items. MSGIFSR designs various granular intent units to model the implicit and multi-granular relationships among items, thus achieving the latest SOTA for session-based recommendation. This suggests the necessity and significance of designing sophisticated modules to capture implicit correlations between items for session-based recommendation.

5.5 Ablation Study (RQ2)

To verify the effectiveness of A-GNN and SG-GNN in DGNN, we removed or replaced one of these modules from DGNN to analyze the performance change.

- **MLP-SR**: replaces A-GNN and SG-GNN with one MLP layer with *ReLU* activation function. The session representation layer, prediction layer and loss function keep the same as DGNN.
- **w/o A-GNN**: removes A-GNN from DGNN.
- **w/o SG-GNN**: removes SG-GNN from DGNN.
- **w/o $\sum$**: removes the accumulated operation in A-GNN and only feeds the output of the latest block into the next block for implicit correlation modeling.
- **w Self-Att**: replaces A-GNN with a multi-head self-attention module [26] and only uses the representation from the last block for implicit correlation modeling.
- **w GGNN**: replaces SG-GNN with the GGNN module in SR-GNN [14, 33].

Our results (Table 3) show that MLP-SR beat all sequential models on all datasets except *LastFM*, which contains much longer sessions than other datasets. The reason lies in that a naive MLP layer could be sufficient for capturing the global information from shorter sessions, while sequential models might be more efficient in handling longer sessions. All modules were shown to be effective, given that removing any of them would drastically degrade the performance.

DGNN outperformed many baselines (e.g., SR-GNN, LESSR, GC-SAN) even without A-GNN, which is impressive, considering SG-GNN only contains half the numbers of parameters and floating point operations (FLOPs) in SG-GNN (refer to Table 4). The performance of DGNN significantly decreased when a self-attention module replaced A-GNN. But changing the information aggregation modules (SG-GNN to GGNN) did not notably impact the results. This indicates that A-GNN is robust to the graph neural networks for explicit dependency modeling as an auxiliary module for session-based recommendation. After removing the accumulated operation in A-GNN (w/o $\sum$), we observed a significant drop in the performance of DGNN. But still, A-GNN beats self-attention even without the accumulation operation. Therefore, we conclude both the output of previous blocks in A-GNN and the *tanh* activation function are critical to our approach.

5.6 Impact of Hyper-parameter Setting (RQ3)

We studied two parameters, the number of A-GNN blocks and the number of SG-GNN layers, in this experiment. Our results (Figure 3) show that a moderate number (neither too large nor too small) of A-GNN blocks generally resulted in better performance. An exception is with the *Yoochoose* dataset, which is extremely large and contains complex transaction patterns. Since the sampled sub-datasets (namely *Yoochoose 1/64* and *Yoochoose 1/4*) cannot fully cover the (original) entire dataset's features, more A-GNN blocks and SG-GNN layers enhance the feature extraction capability to model such complex transaction patterns. Since SG-GNN can obtain the global information from the whole dynamic graph with one convolution operation, a larger number of multiple graph convolution layers will result in smoother item representations, which could negatively affect DGNN's performance.

5.7 Visual Analysis of A-GNN (RQ4)

Session representation. We retrieved two random sessions from the *diginetica* dataset to explore the impact of sequence information on session representation. The two sessions contained the same items in different ordering. We selected a typical sequential neural network, GRU4Rec, as well as our method, DGNN, to generate the session representations. Our results (Figure 4a-b) show that GRU4Rec generated similar representations for session A and session B. These representations, however, significantly differ from the representations output by DGNN. It suggests that sequential

Table 2: Experimental results (%) on the four datasets. The best results are highlighted in boldface, and the second-best results are underlined. * denotes a significant improvement of DGNN over the best baseline results (t-test P<.05).

Model	Diginetica		Yoochoose 1/64		Yoochoose 1/4		Gowalla		Last.FM	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
POP	0.89	0.28	6.71	1.65	1.37	0.31	1.46	0.38	5.26	1.26
Item-KNN	37.75	11.57	51.60	21.81	52.31	21.70	38.60	16.66	14.90	4.04
FPMC	26.53	6.66	45.62	15.01	51.86	17.50	29.91	11.45	12.86	3.78
GRU4Rec	29.45	8.22	60.64	22.89	59.53	22.60	41.98	18.37	17.90	5.39
NARM	49.70	16.00	68.32	28.63	69.73	29.23	50.07	23.92	21.83	7.59
SR-GNN	50.73	17.78	70.57	30.94	71.36	31.89	50.32	24.25	22.33	8.23
SGNN-HN	55.67	19.45	72.13	32.60	73.52	32.63	55.28	27.58	25.07	9.40
LESSR	51.71	18.15	70.59	31.46	72.67	33.12	51.34	25.49	23.37	9.01
MSGIFSR	<u>57.11</u>	<u>20.05</u>	<u>73.13</u>	<u>33.50</u>	<u>74.01</u>	<u>33.74</u>	<u>56.64</u>	<u>29.02</u>	<u>27.63</u>	<u>10.86</u>
GC-SAN	51.70	17.61	70.66	30.04	71.83	30.93	50.68	24.67	22.64	8.42
GCE-GNN	54.02	19.04	70.91	30.63	71.40	31.49	53.96	24.53	24.39	8.63
DGNN	**67.65***	**27.89***	**75.85***	**34.09***	**76.90***	**36.02***	**58.51***	**30.40***	**47.17***	**23.38***
Improv.	18.46%	49.10%	3.72%	1.76%	3.90%	6.76%	3.30%	4.76%	70.72%	115.29%

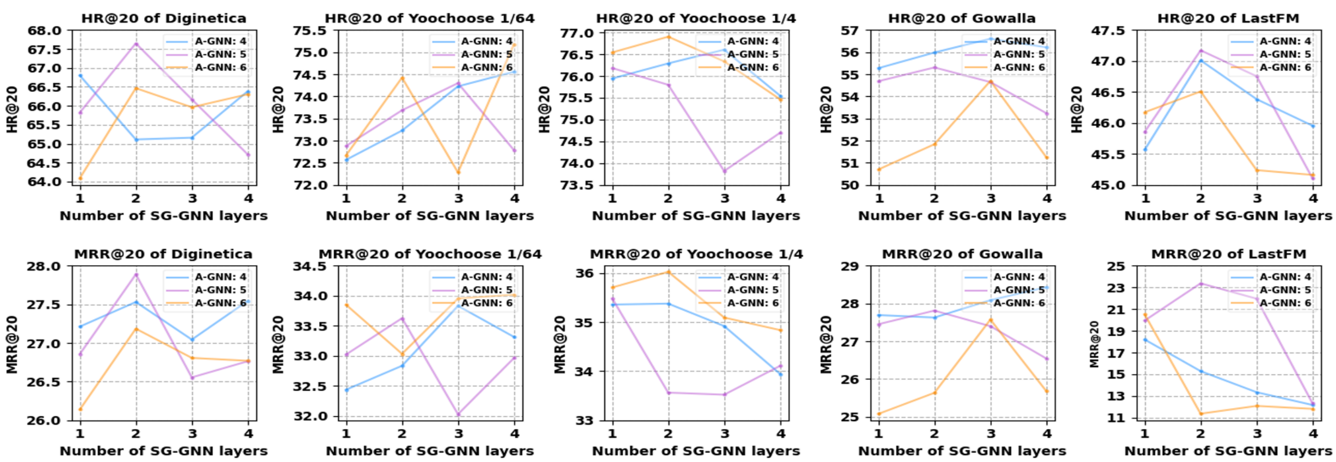


Figure 3: Parameter sensitivity of the number of A-GNN blocks and IP-GNN layers.

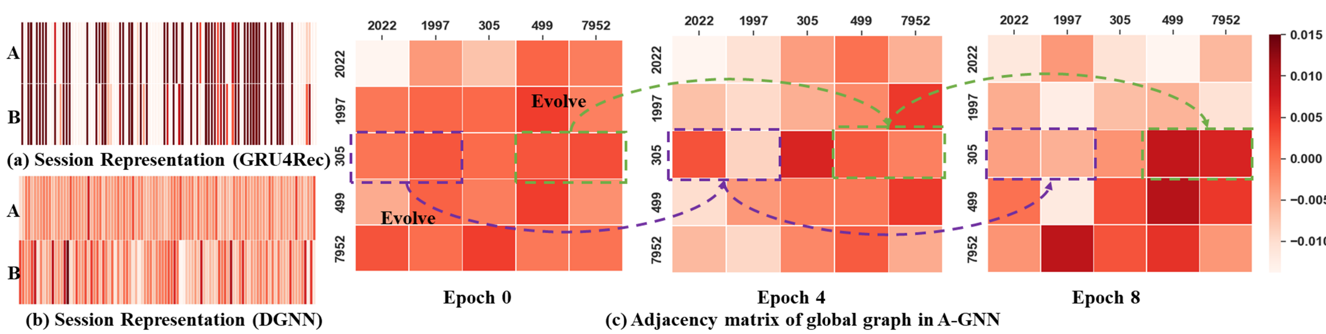
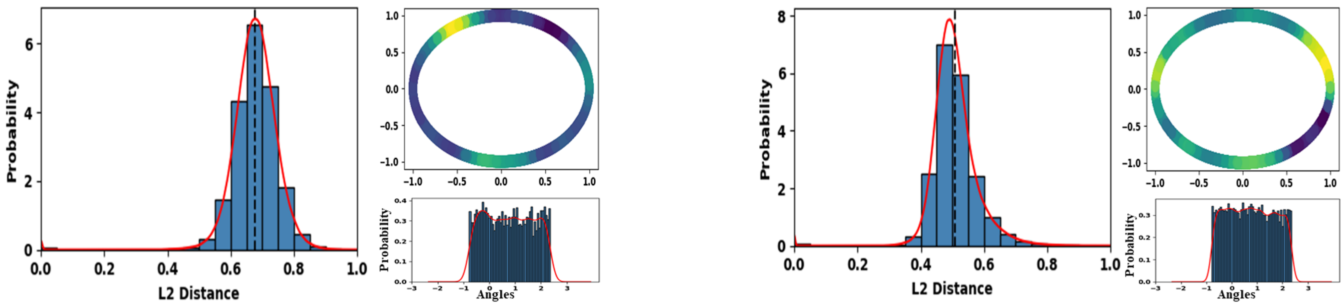


Figure 4: (a) and (b) are the representations of session A:{7951, 7952, 4999, 7952, 305} **and session** B:{4999, 7951, 7952, 305, 7952} **generated by GRU4Rec and DGNN. (c) visualizes the adjacency matrices in A-GNN at epochs 0, 4, and 8, respectively.**

neural networks, e.g., GRU4Rec, may not capture the sequential information in sessions as effectively as our GNN-based approach.

Implicit Correlation. We visualized the global graph adjacency matrices in A-GNN at Epoch 0, Epoch 4, and Epoch 8, to better understand the evolution of implicit correlation between items during the training process. Our results (Figure 4c) revealed that the correlation between any two items was similar at the beginning (Epoch 0). As the training progressed, the correlation between item



(a) Self-attention Module

(b) A-GNN Module

Figure 5: Item representations of (a) self-attention and (b) A-GNN on $\mathcal{S}^1$ (two-dimensional space). Alignment analysis: the histograms show the distributions of l_2 distance between the representations of item pairs, where the black dotted lines indicate the mean distances. Uniformity analysis: the other plots in subfigures show the distributions of item representations with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$ (top-right) and with von Mises-Fisher (vMF) KDE on angles (bottom-right), i.e., $\arctan2(y, x)$ for each point $(x, y) \in \mathcal{S}^1$. The darker the color, the denser the distribution in the top-right plots. Item representations generated by A-GNN are more aligned (lower l_2 distances) and uniform (evenly distributed).

Table 3: Results (%) of ablation experiments.

Datasets		MLP-SR	w/o A-GNN	w/o SG-GNN	w/o $\sum$	w Self-Att	w GGNN	DGNN
Diginetica	HR@20	58.60	53.26	49.67	50.54	50.83	64.22	**67.65**
	MRR@20	20.77	17.71	16.50	17.44	16.54	25.14	**27.89**
Yoo 1/64	HR@20	70.07	71.28	68.10	73.23	70.68	69.25	**75.85**
	MRR@20	30.53	30.87	28.68	32.21	30.84	28.90	**34.09**
Yoo 1/4	HR@20	70.23	74.97	69.88	69.45	75.24	76.64	**76.90**
	MRR@20	31.02	33.41	30.32	30.78	33.38	35.99	**36.02**
Gowalla	HR@20	51.73	52.62	49.47	55.31	50.70	**59.27**	58.51
	MRR@20	25.12	25.63	23.75	26.59	24.37	28.85	**30.40**
LastFM	HR@20	21.98	23.89	21.06	23.39	23.21	38.12	**47.17**
	MRR@20	8.13	9.09	7.74	9.02	8.91	17.47	**23.38**

Table 4: Time and Space Complexity. We set the size of learnable parameter matrices to the dimension of the item embedding d, and the size of graphs to $N \times N$ for SG-GNN and GGNN.

Module	Number of Parameters	FLOPs
GGNN	$d \times (11d + 8)$	$2d \times (2N + 11d)$
SG-GNN	$d \times 5d$	$2d \times (2N + 5d)$

305 (ground truth) and 499, 7952 (previously interacted items) increased while the correlation between 305 and 2022, 1997 (negative samples randomly selected from the item set) dropped steadily. The above results demonstrate that the adjacency matrix in A-GNN can successfully distinguish positive implicit correlations from weak or negative ones between items. This validates our assumption of the existence of implicit correlations between items, with the correlations between items within the same sessions tending to be positive and those between irrelevant items being negative.

Item Representation. We visualized the item representations generated by a self-attention module and A-GNN to offer further insights into the superior effectiveness of A-GNN to the self-attention module in implicit correlation modeling. We consider two key properties in contrastive learning [30] for our visualization task: (1) *alignment* (closeness) of item representations from item pairs; (2) *uniformity* of the induced distribution of the (normalized) item representations on the hypersphere. We randomly selected 5,000 item representations generated by the self-attention module and A-GNN for the *Diginetica* dataset, respectively. Then, we calculated the l_2 distance of any two items to plot the frequency distribution histogram. We further visualized the normalized item representation distribution with Gaussian kernel density estimation (KDE) in $\mathbb{R}^2$. Figure 5 shows the above results. Comparing the l_2 distance distributions of item pairs' representations obtained by the self-attention module (the histogram in Figure 5a) and A-GNN module (the histogram in Figure 5b), we observed that A-GNN resulted in a smaller mean distance (black dotted line) than self-attention, indicating the item representations generated by A-GNN were more closely clustered. A comparison of other diagrams in Figure 5 suggests that A-GNN could obtain a more uniform item representation distribution on $\mathcal{S}^1$. The above analysis implies a uniform and aligned distribution of item representations could benefit session-based recommendation.

6 CONCLUSION

In this paper, we propose to decouple the modeling of explicit dependencies and implicit correlations among items for session-based recommendation. We present a dual graph neural network (DGNN), where a GNN with a single gate (SG-GNN) captures the explicit dependencies as reflected by the ordering of items in sessions, and an adaptive GNN (A-GNN) learns implicit correlations between any two items adaptively with a self-learning strategy. Our extensive experiments demonstrate the superiority of DGNN to SOTA on four public datasets. Besides, A-GNN is shown to generate a more uniform and aligned distribution of item representations.

As we split the batch in a random manner, the adaptive graph will be different in each training process. Thus, the results are somewhat unstable. In future research, we will explore how to construct an effective and efficient adaptive graph for the robust performance.

ACKNOWLEDGMENTS

This work is supported by the Australian Research Council (ARC) under grant No. DP22010371, LE220100078.

REFERENCES

[1] Chen Chen, Jie Guo, and Bin Song. 2021. Dual attention transfer in session-based recommendation with multi-dimensional integration. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 869–878.

[2] Tianwen Chen and Raymond Chi-Wing Wong. 2019. Session-based recommendation with local invariance. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 994–999.

[3] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1172–1180.

[4] Tianwen Chen and Raymond Chi-Wing Wong. 2021. An efficient and effective framework for session-based social recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 400–408.

[5] Wanyu Chen, Fei Cai, Honghui Chen, and Maarten de Rijke. 2019. A dynamic co-attention network for session-based recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1461–1470.

[6] Gabriel de Souza Pereira Moreira, Sara Rabhi, Jeong Min Lee, Ronay Ak, and Even Oldridge. 2021. Transformers4Rec: Bridging the Gap between NLP and Sequential/Session-Based Recommendation. In *Fifteenth ACM Conference on Recommender Systems*. 143–153.

[7] Jiayan Guo, Yaming Yang, Xiangchen Song, Yuan Zhang, Yujing Wang, Jing Bai, and Yan Zhang. 2022. Learning Multi-granularity Consecutive User Intent Unit for Session-based Recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 343–352.

[8] Li He, Hongxu Chen, Dingxian Wang, Shoaib Jameel, Philip S. Yu, and Guandong Xu. 2021. Click-Through Rate Prediction with Multi-Modal Hypergraphs. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 690–699. https://doi.org/10.1145/3459637.3482327

[9] 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*. 843–852.

[10] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[11] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[12] 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.

[13] Yicong Li, Hongxu Chen, Xiangguo Sun, Zhenchao Sun, Lin Li, Lizhen Cui, Philip S. Yu, and Guandong Xu. 2021. Hyperbolic Hypergraphs for Sequential Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong (Eds.). ACM, 988–997. https://doi.org/10.1145/3459637.3482351

[14] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).

[15] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1831–1839.

[16] Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and Maarten de Rijke. 2020. Star graph neural networks for session-based recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1195–1204.

[17] Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 775–783.

[18] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 2019. Rethinking the item order in session-based recommendation with graph neural networks. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 579–588.

[19] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.

[20] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. 2019. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4806–4813.

[21] 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.

[22] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[23] Guy Shani, David Heckerman, Ronen I Brafman, and Craig Boutilier. 2005. An MDP-based recommender system. *Journal of Machine Learning Research* 6, 9 (2005).

[24] Qi Shen, Shixuan Zhu, Yitong Pang, Yiming Zhang, and Zhihua Wei. 2021. Temporal aware Multi-Interest Graph Neural Network For Session-based Recommendation. *arXiv preprint arXiv:2112.15328* (2021).

[25] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.

[26] 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).

[27] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[28] 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*. 345–354.

[29] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian. 2021. A survey on session-based recommender systems. *ACM Computing Surveys (CSUR)* 54, 7 (2021), 1–38.

[30] 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.

[31] Wen Wang, Wei Zhang, Shukai Liu, Qi Liu, Bo Zhang, Leyu Lin, and Hongyuan Zha. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of The Web Conference 2020*. 3056–3062.

[32] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling 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*. 169–178.

[33] 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.

[34] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4503–4511.

[35] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled Side Information Fusion for Sequential Recommendation. *arXiv preprint arXiv:2204.11046* (2022).

[36] 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 *IJCAI*, Vol. 19. 3940–3946.

[37] Haochao Ying, Fuzhen Zhuang, Fuzheng Zhang, Yanchi Liu, Guandong Xu, Xing Xie, Hui Xiong, and Jian Wu. 2018. Sequential recommender system based on hierarchical attention network. In *IJCAI International Joint Conference on Artificial Intelligence*.

[38] Hongyuan Yu, Ting Li, Weichen Yu, Jianguo Li, Yan Huang, Liang Wang, and Alex Liu. 2022. Regularized Graph Structure Learning with Semantic Knowledge for Multi-variates Time-Series Forecasting. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, 2362–2368. https://doi.org/10.24963/ijcai.2022/328 Main Track.

[39] Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang. 2020. Personalized graph neural networks with attention mechanism for session-aware recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[40] Qian Zhang, Shoujin Wang, Wenpeng Lu, Chong Feng, Xueping Peng, and Qingxiang Wang. 2022. Rethinking Adjacent Dependency in Session-Based Recommendations. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 301–313.

[41] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2020. Dgtn: Dual-channel graph transition network for session-based recommendation. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 236–242.