

# Intention Adaptive Graph Neural Network for Category-aware Session-based Recommendation

Chuan Cui<sup>1</sup>, Qi Shen<sup>1</sup>, Shixuan Zhu<sup>1</sup>, Yitong Pang<sup>1</sup>, Yiming Zhang<sup>1</sup>, Zhenwei Dong<sup>1</sup>, and Zhihua Wei<sup>1</sup>

Tongji University, Shanghai, China

{2033065,2130777,2130768,1930796,2030796,1853155,zhihua\_wei}@tongji.edu.cn

**Abstract.** Session-based recommendation (SBR) is proposed to recommend items within short sessions given that user profiles are invisible in various scenarios nowadays, such as e-commerce and short video recommendation. There is a common scenario that user specifies a target category of items as a global filter, however previous SBR settings mainly consider the item sequence and overlook the rich target category information in this scenario. Therefore, we define a new task called Category-aware Session-Based Recommendation (CSBR), focusing on the above scenario, in which the user-specified category can be efficiently utilized by the recommendation system. To address the challenges of the proposed task, we develop a novel method called Intention Adaptive Graph Neural Network (IAGNN), which takes advantage of relationship between items and their categories to achieve an accurate recommendation result. Specifically, we construct a category-aware graph with both item and category nodes to represent the complex transition information in the session. An intention-adaptive graph neural network on the category-aware graph is utilized to capture user intention by transferring the historical interaction information to the user-specified category domain. Extensive experiments on three real-world datasets are conducted to show our IAGNN outperforms the state-of-the-art baselines in the new task.

**Keywords:** Session based Recommendation · Graph Neural Network.

## 1 Introduction

*Recommender systems* (RS) have become indispensable nowadays in scenarios such as online shopping or social media, to provide users with accurate information in an effective way. Thanks to the development in deep neural networks, more and more powerful RS methods have been proposed. Most of them assume the user profile and historical interactions are well recorded. Nevertheless, many services allow user interaction without user identification, therefore *session-based recommendation* (SBR) was proposed specially for this scenario. Most of them are based on *recurrent neural networks* (RNNs) [8,12] or *graph neural networks* (GNNs) [25,18].

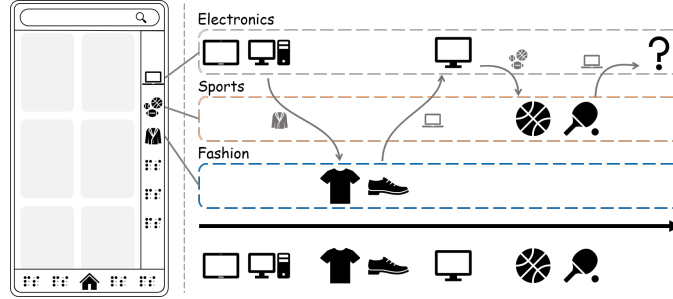


Fig. 1: A toy example of Category-aware Session-based Recommendation.

Previous SBR settings usually focus on the interacted item sequence without consideration of users’ other types of behavior, e.g., clicking the specific category button to filter out items. However, in a real-world SBR scenario, there exist this kind of behavior beyond the interactions with items. For instance, as shown in Figure 1, it illustrates a regular page that supports specifying one category and filtering items in other categories, with the selection button on the navigation bar. During an online shopping trip, after several browsing among different categories, user may wonder if there is any interesting item in the category “Electronics”. Therefore he taps the “Electronics” button and interacts with items in this target category. This can happen when user has the following requirements: (i) The recommendation direction deviates from the user’s intention, therefore user wants to adjust the recommendation results via additional feedback in time, e.g., specifying the target category in accord with his intention. (ii) Sometimes the recommendation results could not keep up with user’s dynamic and abruptly shifting intention, therefore he may want to filter the items with target category to locate the right item efficiently. (iii) During the interaction, user may tend to concentrate on items in one category than the others, and an autonomous category selection by the user becomes necessary. In summary, these scenarios require the system to model the dynamic and rich interaction sequence (including interactions with item and category), for making more accurate and user-controllable recommendations based on user-specified target category, rather than general recommendations under all categories. When facing above scenarios, previous SBR settings cannot perceive the specific category which user intends to view next, and the system fails in reacting to the shifted interest promptly, leading to the failure of meeting user expectations.

Due to limitations listed above, we propose a new task for SBR: In an ongoing session, with user-specified target category and each item’s corresponding category, the system recommends the next item specifically in the target category. The category selection can be commonly seen in the navigation bar of online applications, e.g., the page of an e-commerce platform in Figure 1. This task, namely Category-aware Session-Based Recommendation (CSBR), extends the interaction mode of users, *w.r.t.* specifying target category of next items, to accomplish a user-controllable SBR.

As for the SBR with target category, the key challenge is how to efficiently leverage the category information in addition to the original interaction sequence. A straightforward adaption method is to recommend the next item based on the interaction history via existing SBR models and then filter the results by the target category. However, this approach overlooks something which are the challenges for CSBR listed as follows: (i) How to inject the auxiliary category information into session representation dynamically. Compared with previous SBR methods, the additional categories information need to be further considered, including category-level user interaction transition and item-category relations. (ii) How to transfer historical interaction information to the user-specified category domain efficiently. Intuitively, the interaction in one category might be helpful to improve recommendation effectiveness in other categories, for the idea that user behavior in different categories might reveal a particular user characteristics or interest. For example, as shown in Figure 1, a user might firstly interact with several items in the order of categories “Electornics”, “Fashion”, “Electronics”, “Sports” and then he clicks the “Electronics” category button to get the recommendation results only in “Electronics” category. During this process, some fashion item like “T-shirt with Super Mario print” may be helpful to predict the next item “Nintendo console” in the “Electronics” category because they are all related to the video game “Mario” series. However, adopting user interaction information from other categories will incorporate helpful but also noisy information at the same time. For instance, a “Swimming google” in “Sports” category might be irrelevant to a game console. Therefore, it is crucial to distill intention-adaptive information for matching the current interest of user.

To effectively address the aforementioned challenges, we propose a novel model named Intention Adaptive Graph Neural Network (IAGNN) for CSBR. In detail, we start by converting the complex interaction session into an Category-aware Graph, which is composed of item-level transitions and category-level transitions, and item-category relations. Besides, to transfer historical interaction information to the user-specified category domain, we instantiate the message path between target category node and item nodes with explicit links in the graph. Based on the lossless graph, we employ a position-aware graph neural network to learn the item-level and category-level representations. By stacking multiple layers, the complicated historical interaction information is aggregated to refine the user intention representation in target category iteratively. Finally, the user’s intention is characterized by the target category and last item representations for CSBR.

Our main contributions of this work are summarized below:

- We consider a flexible session-based recommendation scenario when user’s autonomous navigation in possible item category is perceived, which is introduced as the task **CSBR**, concentrating on leveraging the signals of user-specified target category and item interaction sequence, for more precise recommendation results.

- To address the challenges of CSBR, we propose a novel model named Intention Adaptive Graph Neural Network (**IAGNN**). Firstly, we construct a category-aware graph not only explicitly models the historical transitions and item-category relations, but also covers the connections of user-specified target category and historical interacted items. Moreover, we conduct a position-aware intention-adaptive graph neural network on the category-aware graph to capture user intention by transferring the historical interaction information to the user-specified category domain.
- Extensive experiments on three datasets demonstrate that our model is superior compared with state-of-the-art models for CSBR.

## 2 Related Works

In this section, we review some related works of session-based recommendation and category-based recommendation.

### 2.1 Session-based Recommendation

Various neural network based approaches have been proposed for session-based recommendation(SBR) with the development of deep learning recently.

**RNN-based SBR.** Recurrent neural networks (RNNs) are powerful sequence models which have been widely adopted for SBR tasks [8,12,14,27,21,23]. GRU4Rec [8] by Hidasi et al. was the first RNN-based method to capture information in user-item interaction sequences by simply utilizing several GRU layers. NARM [12] takes advantage of attention mechanism beyond GRU4Rec, by referring the last interacted item, and captures user’s preference representations both from global and local perspective of current session. However, these sequential methods are unable to capture the items transition relationship efficiently.

**GNN-based SBR.** Recently, graph neural networks (GNNs) has been proved to be competent to extract complex relationships between objects, so there emerges quite a few GNN-based methods for SBR [25,26,20,24,19,22]. SRGNN [25] is the first work which leverages gated GNN (GGNN) on a directed graph constructed from the interaction sequence to learn item embeddings. Nevertheless, SR-GNN only propagates messages between adjacent items, which would fail to take long-distance item relations into consideration. LESSR [4] proposes a better architecture, including EOPA layer and SGAT layer to solve the lossy session encoding problem and propagate information along shortcut connections, which leads to lossless information presentation during graph construction and better performance for SBR. Pan et al. proposed StarGNN [18] with a star graph neural network to model the complex transition relationship between items with an additional star node connected to every item in an on going session, and applies a highway network to handle the over-fitting issue in GNNs.

### 2.2 Category information in Recommendation

As a significant auxiliary information for items, category information has been explored in other recommendation areas.

**Cross-domain recommendation.** Cross-domain recommendation (CDR) [9,10] can be considered as general multi-category recommendation, which utilizes data from multiple domains to deal with issues like cold start [1] and data sparsity [17] in target domain (category). Recently, Ma et al. proposed  $\pi$ -net [15] to generate recommendation scores for every item in two domains by integrating information from both. DA-GCN [6] constructs a cross-domain sequence graph which explicitly links account-sharing users and items from two domains to learn expressive representations for recommendation.

**Category-aware recommendation.** Category-aware recommendation utilizes category information to enhance the item representation for better user preference modeling. There are several traditional category-aware recommendation methods. HRPDA [2] proposes a hybrid recommendation method to handle the customer preferences varieties in different product categories. Choi et al. [5] design a recommendation algorithm based on the category correlations to a user with certain preferences. Recently, LBPR [7] conducts list-wise bayesian ranking for next category and category-based location recommendation. CoCoRec [3] utilizes self-attention to model item transition patterns in category-specific action sub-sequences, and recommends items for user by collaborating neighbors' in-category preferences.

As outlined above, the rich category information has not been comprehensively explored by previous SBR methods. Also for previous category-related methods, they rarely focused on recommendation for a session with anonymous users. What is more, none of these task settings explicitly consider the user-specified category information, and there are huge gaps in this scenario.

### 3 Preliminary

Given the entire item set  $\mathcal{V}$  and category set  $\mathcal{C}$ , we first define a category-augmented item session as  $s = \{(v_1, c_{v_1}), (v_2, c_{v_2}), \dots, (v_n, c_{v_n})\}$ , in which  $(v_i, c_{v_i})$  represents the user interacted item  $v_i \in \mathcal{V}$  in category  $c_{v_i} \in \mathcal{C}$ , and  $n$  is the session length. Note that  $v_i$  and  $c_{v_i}$  can be repetitive in the sequence, since there can be repeated items in the session. Given session  $s$ , the user-specified target category  $c_t$  and the corresponding item set  $\mathcal{V}_{c_t} = \{v_i | c_{v_i} = c_t, v_i \in \mathcal{V}\}$ , the goal of CSBR is to predict a probability score for any item  $v \in \mathcal{V}_{c_t}$  such that an item with higher score is more likely to be interacted next.

## 4 Methodology

### 4.1 Overall Architecture

Figure 2 illustrates the overall architecture of IAGNN under the context of user's target category specified. First, the Embedding Layer will initialize id embeddings for all items and categories. Second, we construct the Category-aware Graph to explicitly keep the transitions of in-category items and different categories, along with the relation between items and their corresponding categories.

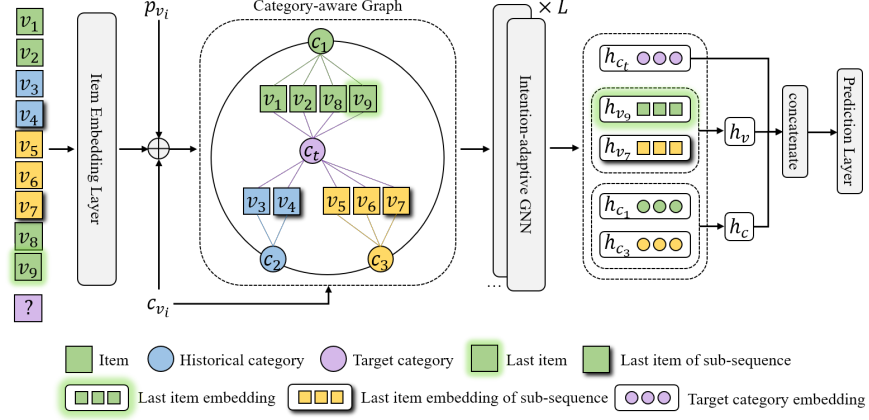


Fig. 2: The overview of IAGNN.

Third, a Heterogeneous Graph Attention Neural Network is introduced to propagate embeddings through attentive graph convolution. Finally, by leveraging a Embedding Fusion Layer and a Prediction Layer, items in user-specified target category will be recommended.

## 4.2 Category-aware Graph Construction

According to previous SBR methods, category information is not explicitly integrated in the recommendation process. In our proposal, we take advantage of graph neural network to model the items and categories transition. The graph  $\mathcal{G}_s = (\mathcal{V}_s, \mathcal{E}_s)$  is a directed heterogeneous graph, in which  $\mathcal{V}_s = (\{v_1, v_2, v_3, \dots, v_n\}, \{c_1, c_2, c_3, \dots, c_h\}, c_t)$  indicates  $n$  item nodes,  $h$  category nodes and the user-specified target node  $c_t$  in the graph.  $\mathcal{E}_s$  is comprised of the edges representing item transitions, category transitions, item-category connections and target category connections. Therefore, the semantics of item transition and category transition can be represented by these nodes and their relations. Generally, there are four sub-graphs with independent edge relations, and we will detail them as follows.

**Item Transitions.** At first, we split the original interaction sequence into several category-specific item sub-sequences. For instance, items in category  $c$  will be connected with their original interaction order in the session, and the same for other categories. Therefore, the in-category item-to-item transition patterns can be modeled in the sub-sequence.

In detail, for each item node  $v_i$  in the category  $c_{v_i}$ , we create a link starting from  $v_i$  to its next item  $v_j$ , representing the item transition in the category sub-sequence. And we also create self-loop for each item node to include the node itself when aggregating.

**Category Transitions.** Considering each item node  $v_i$  in the original session sequence, we can simultaneously initialize a corresponding category node  $c_{v_i}$  for it. And a category transition graph can be constructed according to the

order of the original session sequence order. For instance, in Figure 2,  $c_1$ ,  $c_2$ ,  $c_3$  indicate the categories in colors ‘Green’, ‘Blue’ and ‘Yellow’. The session sequence consists of items in category  $[c_1, c_2, c_3, c_1]$  respectively, then a sub-graph composed of connections between categories can be constructed by the directed link  $c_1 \rightarrow c_2 \rightarrow c_3 \rightarrow c_1$ . Similar to the item transition sub-graph, a self-loop is added to every category node  $c_i$ .

**Item-Category Connections.** For representing the inherent item-category relations, we build bi-directed connections between item node  $v_i$  and its corresponding category node  $c_{v_i}$ . By this connection, information from non-adjacent items nodes can be propagated in a two-hop way through the corresponding category node as an intermediate node. Besides, the category transition information which implies the user interest changes across different categories can be propagated back to all the related item nodes. Finally, this approach can enhance the representation of both items and categories.

**Target Category connections.** Moreover, the user-specified target category is a given information in our task, to better leverage this information, we add a particular node  $c_t$  to represent the target category, which is connected to all the item nodes (In Figure 2,  $c_t$  represents the same category with  $c_3$ ). Therefore, it can not only solve the long-range information propagation issue between non-adjacent items, but also build the transfer from all the historical information to the target category domain as the the process of message passing from items to target category node. Meanwhile, as a pseudo node for the next item, the representation of  $c_t$  can be an intermediate variable to bridge the gap of all the interacted items and the actual next item in the session.

### 4.3 Intention-adaptive Graph Neural Network

Next, we present how intention-adaptive graph neural networks propagate messages between different nodes.

**Embeddings.** For each item  $v_i$ , it is projected by the item embedding layer  $E_i \in \mathbb{R}^{|V| \times d_v}$  into dense embedding  $e_{v_i}$ . In which  $d_v$  denotes the item embedding size. Meanwhile the category embedding layer  $E_c \in \mathbb{R}^{|C| \times d_c}$  transform category unique identifications to category embedding  $e_{c_i}$ . Note that since  $c_t$  represents the target category, it can also be initialized in the same way as a regular category node.

Also for a session with no more than  $n$  items, we take advantage of the reversed position embeddings  $\mathbf{P}_{-i} \in \mathbb{R}^{n \times d_P}$  by following GCE-GNN[24], which shows superiority performance because the session length is versatile. Moreover, we conduct a positive-going position embedding in our experiments. Also the category embeddings  $\mathbf{e}_{c_{v_i}}$  are added for each item  $v_i$  to enhance the item representation by concatenation and projection:

$$\mathbf{e}_{v_i} = \mathbf{W}' [\mathbf{e}_{v_i} \parallel \mathbf{e}_{c_{v_i}} \parallel \mathbf{p}_{v_i}] \quad (1)$$

where  $\parallel$  is the concatenation operation,  $\mathbf{W}' \in \mathbb{R}^{d_v \times (d_v + d_c + d_P)}$  is the projection matrix to keep the size of  $e_{v_i}$ ,  $\mathbf{p}_{v_i} \in \mathbf{P}_{-i}$  is the reversed position embedding for  $v_i$ .

**Message propagation.** Our method uses the same message propagation mechanism for the different edge types in the graph. We use item-item message propagation as an example here. As mentioned above, the item node  $v_i$  is linked to its previous item in the in-category sequence along with the self-loop connection. Then the item message that node  $v_i$  received can be represented as:

$$\mathbf{h}_{v_i \leftarrow v}^{(l+1)} = \sigma \left( \sum_{j \in \mathcal{N}_{v_i}(v)} \alpha_{ij}^{(l)} \mathbf{W}_v \mathbf{h}_{v_j}^{(l)} \right) \quad (2)$$

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\mathbf{a}_{l_v} \mathbf{W}_v \mathbf{h}_i + \mathbf{a}_{r_v} \mathbf{W}_v \mathbf{h}_j))}{\sum_{k \in \mathcal{N}_{v_i}(v)} \exp(\text{LeakyReLU}(\mathbf{a}_{l_v} \mathbf{W}_v \mathbf{h}_i + \mathbf{a}_{r_v} \mathbf{W}_v \mathbf{h}_k))} \quad (3)$$

where  $\mathbf{h}_{v_i \leftarrow v}^{(l+1)} \in \mathbb{R}^{d_v}$  is the item-source message representation of node  $v_i$  at layer  $(l+1)$  and  $\mathcal{N}_{v_i}(v)$  stands for the set of neighbor item nodes.  $\alpha_{ij}$  is a scalar representing the attention coefficient.  $\mathbf{W}_v \in \mathbb{R}^{d_v \times d_v}$  and  $\mathbf{a}_{l_v}, \mathbf{a}_{r_v} \in \mathbb{R}^{1 \times d_v}$  are the shared linear transformation parameters for the source and target node.

**Graph nodes aggregation.** Similar to the item-item message aggregation, we can get other types of messages from the adjacent nodes. Therefore, the final representation for an item node at layer  $(l+1)$  can be represented as:

$$\mathbf{h}_{v_i}^{(l+1)} = \mathbf{h}_{v_i \leftarrow v}^{(l+1)} + \mathbf{h}_{v_i \leftarrow c}^{(l+1)} + \mathbf{h}_{v_i \leftarrow c_t}^{(l+1)} \quad (4)$$

where  $\mathbf{h}_{v_i \leftarrow v}$  denotes the messages passed from other item nodes,  $\mathbf{h}_{v_i \leftarrow c}$  represents the messages from its corresponding category node, and  $\mathbf{h}_{v_i \leftarrow c_t}$  stands for the messages from target category.

As for the category nodes, its representation at layer  $(l+1)$  is:

$$\mathbf{h}_{c_i}^{(l+1)} = \mathbf{h}_{c_i \leftarrow v}^{(l+1)} + \mathbf{h}_{c_i \leftarrow c}^{(l+1)} \quad (5)$$

In which  $\mathbf{h}_{c_i \leftarrow v}^{(l+1)}$  is the messages propagated from adjacent item nodes, and  $\mathbf{h}_{c_i \leftarrow c}^{(l+1)}$  represents the messages from adjacent category nodes.

The same for the target category node  $c_t$ , its  $(l+1)$  layer representation is:

$$\mathbf{h}_{c_t}^{(l+1)} = \mathbf{h}_{c_t \leftarrow v}^{(l+1)} \quad (6)$$

Since the target category node connects to all the item nodes, only  $\mathbf{h}_{c_t \leftarrow v}^{(l+1)}$ , the messages aggregated from item nodes are used for its representation.

As we employs an  $L$ -layer IAGNN, the embeddings for each node before and after the IAGNN can be represented as  $\mathbf{h}^{(0)}$  and  $\mathbf{h}^{(L)}$  respectively. In order to fuse the semantics before and after the network and avoid possible overfitting issue, we utilize the gate mechanism to get the final embeddings for each kind of node:

$$\mathbf{h}^L = \mathbf{g} \odot \mathbf{h}^{(0)} + (1 - \mathbf{g}) \odot \mathbf{h}^{(L)} \quad (7)$$

$$\mathbf{g} = \sigma(\mathbf{W}[\mathbf{h}^{(0)} \parallel \mathbf{h}^{(L)}]) \quad (8)$$

where the  $\mathbf{g}$  is the gate factor to control information contributed from  $\mathbf{h}^{(0)}$  and  $\mathbf{h}^{(L)}$ .  $\mathbf{W} \in \mathbb{R}^{(d_h \times 2) \times d_h}$  are the transform matrix to get the gate.  $\sigma$  denotes the *sigmoid* activation function.



#### 4.4 Embeddings fusion and Prediction

In a SBR scenario, the last item user interacted contributes a lot to the next item user will interact with. In our case, there are two last items, one is the last item  $v_l$  of the original interaction sequence, the other is the last item  $v_{l_c}$  of the in-category sub-sequence. Therefore, after going through the IAGNN, we fuse these two item representations to a session item representation:

$$\mathbf{h}_v = \text{Gated}(\mathbf{h}_{v_l}^L, \mathbf{h}_{v_{l_c}}^L) \quad (9)$$

Accordingly, we fuse the category node embeddings  $c_l$  and  $c_{l_c}$  which is related to the sequence last item  $v_l$  and last item  $v_{l_c}$  of in-category sub-sequence as the session category representation:

$$\mathbf{h}_c = \text{Gated}(\mathbf{h}_{c_l}^L, \mathbf{h}_{c_{l_c}}^L) \quad (10)$$

After obtaining the item and category session representation  $\mathbf{h}_v$  and  $\mathbf{h}_c$ , we can also get the embedding of target category node:

$$\mathbf{h}_{c_t} = \mathbf{h}_{c_t}^L \quad (11)$$

Then, we combine them into one embedding for the session:

$$\mathbf{h}_s = \mathbf{W}_s [\mathbf{h}_v \parallel \mathbf{h}_c \parallel \mathbf{h}_{c_t}] \quad (12)$$

where  $\mathbf{W}_s \in \mathbb{R}^{d_v \times (d_v + d_c \times 2)}$  is used to project the concatenation result to an embedding of size  $d_v$ . Note that we did not introduce additional attention mechanism for all the items as a readout step, because regarding the target category node, the attention mechanism is equivalently done after message propagation through multi-layer GNN.

By multiplying the session representation with all item embeddings, we can get the prediction score as follows:

$$\hat{y}_i = \mathbf{h}_s^\top \mathbf{e}_{v_i} \quad (13)$$

where  $\mathbf{e}_{v_i}$  is the projected candidate item embeddings through item embedding layer  $E_i \in \mathbb{R}^{|V| \times d_v}$ .

To train our network, a cross-entropy loss function is employed to optimize the model parameters:

$$L(\hat{\mathbf{y}}) = - \sum_{i=1}^{|V|} y_i \log(\hat{y}_i) + (1 - y_i) \log(\hat{y}_i) \quad (14)$$

where  $y_i$  denotes the ground-truth item one-hot encoding vector.

## 5 Experiments

We conduct extensive experiments to evaluate our method in comparison with other state-of-the-art methods<sup>1</sup>. The goal in this section is to answer the following research questions:

- **RQ1:** What is the performance difference between SOTA baselines and ours?

<sup>1</sup> Source code and data will be released in the future for research purpose.

- **RQ2:** Can the category information of the sequence items contribute to the recommendation performance?
- **RQ3:** Will the graph construction benefit the model performance?
- **RQ4:** How does our model perform on the task CSBR with different hyper-parameters setup?
- **RQ5:** How do the models perform on the session with different target category frequencies?

Table 1: Statistics of datasets used in experiments.

Statistic	Diginetica*	Yoochoose*	Jdata*
No. of items	33,596	13,317	79,356
No. of sessions	482,743	400,613	959,824
Avg. of session length	6.48	10.27	6.90
No. of categories	982	12	79
Avg. of categories per session	2.28	2.07	2.35

### 5.1 Experimental Setup

**Dataset.** We performed the evaluation on three public datasets: Diginetica\*, Yoochoose\* and Jdata\*, which are widely used in the session-based recommendation research [4,24,16]. These datasets contain additional category information which can support our work for CSBR.

- Diginetica\*<sup>2</sup> includes user sessions extracted from e-commerce search logs, with desensitized user ids, hashed queries, hashed query terms, hashed product descriptions and meta-data, log-scaled prices, clicks, and purchases.
- Yoochoose\*<sup>3</sup> is the dataset for RecSys Challenge 2015, which contains user clicks and purchases of an online retailer within several months. In our case, we use the fractions 1/4 of Yoochoose data.
- Jdata\*<sup>4</sup> is also a dataset of a challenge hosted by JD.com. The data was filtered by 1 hour duration to extract the session data.

To better apply the datasets to our task, we pre-process them before training. At first, as described in [12,14,25], we applied a data augmentation by regarding the  $i$ th as the label and items before  $i$  as the input sequence. Moreover, the category of the label item is considered as the user-specified target category. Then, since there could be short sessions or infrequent items, we removed all sessions of length  $\leq 2$  and items which have an occurrence less than 5 times in all datasets. In the end, for the given specific category in one session, we filtered the candidate items by keeping the category of candidate item the same as the target category. We summarized the preprocessed dataset in Table 1.

<sup>2</sup> <https://competitions.codalab.org/competitions/11161>

<sup>3</sup> <https://www.kaggle.com/chadgostopp/recsys-challenge-2015>

<sup>4</sup> <https://jdata.jd.com/html/detail.html?id=8>

Table 2: Experimental results (%) of different models in terms of  $P@10, 20$ , and  $mrr@10, 20$  on three datasets. The \* means the best results on baseline methods. *Improv.* means improvement over the state-of-art methods. The bold number indicates the improvements over the strongest baseline are statistically significant ( $p < 0.01$ ) with paired t-tests.

Models	Diginetica*				Yoochoose*				Jdata*			
	P@10	P@20	mrr@10	mrr@20	P@10	P@20	mrr@10	mrr@20	P@10	P@20	mrr@10	mrr@20
GRU4Rec	27.47	36.52	15.02	15.89	10.25	13.93	3.92	4.38	31.20	43.79	13.91	14.97
NARM	28.86	37.84	16.89	17.53	11.86	15.25	5.98	6.21	33.92	45.01	15.52	16.29
SRGNN	30.71	41.86	16.27	17.17	11.46	15.39	4.84	5.11	35.00	46.30	16.07	16.88
LESSR	30.49	41.42	16.52	17.27	11.90	15.67	5.85	6.24	35.95	46.92	16.95	17.71
StarGNN	29.86	40.96	15.33	16.09	12.46	17.30	5.47	5.80	36.91	48.40	17.15	17.94
NARM <sup>†</sup>	28.79	38.48	16.95*	17.61	11.95	15.76	6.28	6.54	33.98	44.97	15.61	16.38
LESSR <sup>†</sup>	30.55	41.39	16.77	17.51	12.95	17.24	6.20	6.52	36.05	47.22	17.01	17.79
StarGNN <sup>†</sup>	30.67	41.34	16.93	17.66*	11.73	16.27	5.24	5.55	37.21*	48.75*	17.53*	18.33*
DA-GCN <sup>†</sup>	31.49*	42.53*	16.86	17.62	15.53*	20.15*	7.09*	7.41*	36.42	48.30	16.52	17.29
IA-GNN	<b>32.63</b>	<b>43.80</b>	<b>17.35</b>	<b>18.11</b>	<b>17.03</b>	<b>21.42</b>	<b>8.18</b>	<b>8.49</b>	<b>39.86</b>	<b>51.79</b>	<b>18.89</b>	<b>19.72</b>
<i>Improv.</i>	3.62%	2.99%	2.36%	2.55%	9.66%	6.30%	15.37%	14.57%	7.12%	6.24%	7.76%	7.58%

**Baseline Models.** To evaluate the performance of our method, we compare it with several baselines. Note that the candidate items in the experiments are all filtered with the target category, which means we evaluate the precision and rank performance only on the candidate items in the target category for all models.

- **GRU4Rec** [8] captures patterns in user-item interaction sequences by simply utilizing several GRU layers.
- **NARM** [12] combines the global and local representations to generate the session embedding through attention mechanism and RNN model.
- **SRGNN** [25] transforms user interaction sequences into directed graphs and utilizes the GGNN layer [13] to learn the embeddings of sessions and items.
- **LESSR** [4] uses a lossless GRU and a shortcut graph attention layer to capture long-range dependencies and lossless embeddings.
- **StarGNN** [18] employs a additional star graph neural network to model the complex transition in sessions.
- **DA-GCN<sup>†</sup>** [6] uses a domain-aware GNN and two novel attention mechanisms to learn the sequence representation. Here we change the domain concept of the original DA-GCN to category in DA-GCN<sup>†</sup> by employing in-category sub-sequences, along with the target category integrated.

Nevertheless, we adopt the target category information to NARM, LESSR, StarGNN by fusing the target category embedding to the query part of their attention mechanism, named NARM<sup>†</sup>, LESSR<sup>†</sup> and StarGNN<sup>†</sup>.

**Evaluation Metrics.** We employ two widely used metrics: *Precision* ( $P@k$ ) and *Mean reciprocal rank* ( $mrr@k$ ) following [25, 4], where  $k = \{10, 20\}$ . They respectively represent the correct proportion of the top- $k$  result items and the order of recommendation ranking.

**Implementation Details.** By leveraging the frameworks PyTorch and Deep Graph Library, we implement our method. We fix the embedding size of items and categories to 128. The model parameters are initialized with a Gaussian distribution with  $\mu = 0$  and  $\sigma = 0.1$ , where  $\mu$  and  $\sigma$  are the statistical mean and standard deviation. To help the model converge, We employ the *Adam*[11] optimizer with the mini-batch size of 256. A grid search for the hyper-parameters is the following: learning rate  $\eta$  in  $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ , learning rate decay step in  $\{2, 3, 4\}$ , number of graph neural network layers  $L$  in  $\{1, 2, 3, 4, 5\}$ . Furthermore, we split the dataset to train, test and validation set by ratio 8:1:1. Regarding the baseline methods, we either directly make minimal changes to their original source code for supporting our dataset, or implement by ourselves according to their papers.

## 5.2 Experimental Results (RQ1)

Table 2 shows the comparison results of IA-GNN over other baselines on the preprocessed datasets *Diginetica*<sup>\*</sup>, *Yoochoose*<sup>\*</sup> and *Jdata*<sup>\*</sup>.

**Comparison of Different Baselines.** The performances of two RNN-based methods, GRU4Rec and NARM, are not so competitive. Nevertheless, NARM performs much better than GRU4Rec because of its usage of attention mechanism to capture the user interest. And for the GNN-based methods SRGNN, LESSR and StarGNN, they performs significantly better than the RNN-based methods, which proved that the GNN has promising advantage for SBR. Moreover, we can discover that LESSR and StarGNN outperform SRGNN in most cases because they use shortcut graph or the star node to capture global dependencies between distant items, showing the effectiveness of explicitly considering long-range dependencies.

**Significance of Target Category.** Furthermore, for the methods with target category information adopted, NARM<sup>†</sup>, LESSR<sup>†</sup> and StarGNN<sup>†</sup> generally outperform their original implementation, which indicates that integrating user-specified target category information can improve the performance significantly for CSBR. We can notice that DA-GCN<sup>†</sup> performs better than the other modified baselines. This means the graph construction method including in-category sub-sequence can benefit our task. Note that for dataset *Diginetica*<sup>\*</sup>, the overall improvements are not obvious because it has too many categories which can be noisy to the category enhanced models.

**Model Effectiveness.** Our model outperforms all the other GNN-based models from Table 2, which shows our model has a better effectiveness. We can elaborate the advantage of our method in three aspects. First, we integrate the user-specified target category into the graph as a dedicated node, such that the model can be aware of the target category as the context of the session. Second, we inherit the in-category sub-sequence graph construction from the CDR methods, which can model the item transition per category. Third, we introduce the nodes and transitions for categories, and along with the in-category sub-sequence, we keep the original item sequence which can lead to less information loss.

Table 3: Performance comparison for ablation study.

Model setting	Yoochoose*		Jdata*	
	P@20	mrr@20	P@20	mrr@20
w/o Category nodes	20.89	8.12	50.84	19.32
w/o Target category information	20.91	8.04	50.87	19.08
w/o Category transition	21.36	8.13	51.14	19.47
Add Original item transition	21.22	8.41	51.70	19.68
Add Attention mechanism	20.76	7.68	50.14	18.97
Positive position information	21.80	8.04	51.49	19.62
IAGNN	21.42	8.49	51.79	19.72

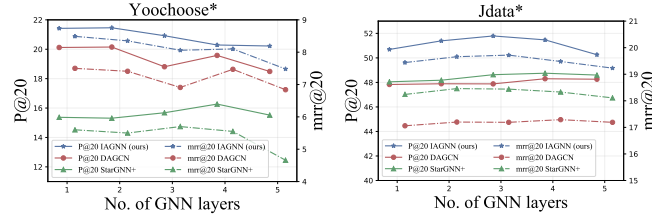


Fig. 3: Model comparison w.r.t. different depths of GNN.

### 5.3 Ablation Study (RQ2&3)

In this section, we performed some ablation studies to demonstrate the effectiveness of our model designs.

**Category information.** In our proposed method, we consider the user-specified target category as a dedicated node and other category nodes for in-category sub-sequences during the graph construction. Therefore, we compared our model with the version without a target category node (“w/o Target category node”) or sub-sequence category nodes (“w/o Category nodes”) in the graph in order to show the effectiveness of introducing these nodes. On the other hand, to show the importance of introducing user-specified target category information, we used mean pooling of all item representations as the representation of the target category node instead of the target category (“w/o Target category information”) embedding to diminish the impact of target category for comparison. Moreover, we compared the case when transitions between each category node were removed (“w/o Category transition”). As illustrated in Table 3, the performance dropped on both datasets while detaching category information from the model in different mechanisms. This proved the graph construction details aforementioned in section 4 help improving the model performance.

**Additional model operations.** As we know, the item sequence in session has its inherent order relation, and we added the reversed position embedding for every item node to keep the original item transition information. Thus we tried to add the original item transition as directed links for all the item nodes to check if an additional lossless item order can improve the performance. Meanwhile, following [12,25,4], an additional attention mechanism which gets each

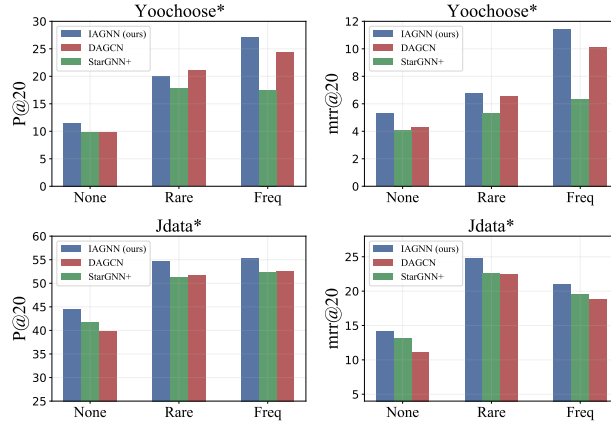


Fig. 4: Model comparison w.r.t. different groups.

item contribution was added to our model. Also we want to see if a positive position embedding is better than the negative. Corresponding to above three conditions, we respectively conduct three alternative models: “Add Original item transition”, “Add Attention mechanism” and “Positive position information” for comparison. Following the result in Table 3, we can observe that by adding original item transition links to the method, the performance drops a little. The reason is the model already takes advantage of the negative position embedding and category transition which persist the original item transition order, adding these links will be redundant for the model. Meanwhile, by introducing attention mechanism, the performance was impaired because the target category information is much more significant than attention from every item which is in different categories. About the position information, albeit positive position embedding improves for P@20 on dataset Yoochoose\*, overall the performance with this information drops comparing to our method with negative position embedding.

#### 5.4 Hyper-parameters Study (RQ4)

**GNN depths.** We conducted experiments on different number of GNN layers for different GNN-based methods to check the performance impact. As illustrated in Figure 3, our method keeps beyond the others with different GNN layers. Also, we notice that the performance goes up when multi-layer GNN was conducted than that of only one single layer, because more GNN layers are able to capture the high-level and complex semantic information. Furthermore, the metrics of all methods decline approximately after 4 GNN layers, which indicates the over-smoothing is a common issue for GNN-based methods. Without exception, the performance of our method drops after about 3 layers. Nevertheless, as the number of GNN layers goes up, the performance of almost every method increases at the beginning, because GNN is competent for capturing the node embedding transitions, showing it is suitable for our task which involves item and category transitions.

### 5.5 Analysis on Different Target Category frequencies (RQ5)

We further analyze the capability of different models to cope with sessions of different target category frequencies. For comparison, we partition sessions of Yoochoose\* and Jdata\* into three groups based on occurrence number of the target category in current session, where “None” indicates that a new category never interacted, and “Rare” means that the frequency is less than 3, while each session has more than 3 target categories in “Freq”. The percentages of session belonging to “None”, “Rare” and “Freq” group are  $\{0.28 : 0.42 : 0.30\}$  and  $\{0.29 : 0.44 : 0.27\}$  on the Yoochoose\* and Jdata\*, separately. As shown in Figure 4, our model outperforms other competitive baselines on two datasets with different target category frequencies. It demonstrates the superior performance of IAGNN. Meanwhile, all models achieve good performance on the “Freq” group, and the performance drops with the frequency decreasing. This is partially because more item interactions in target category will reveal more effective information of the user interest in target category. Especially, this task degenerates into an approximate cold-start problem for the “Rare” group without the item record in target category, and the result is worse.

## 6 Conclusion

This paper proposes a novel category-aware session-based recommendation task, which includes the rich category information both resided in the interaction sequence and the user’s intention. Accordingly, a novel model Intention Adaptive Graph Neural Network is introduced to take advantage of these category information, and achieves a more effective performance by explicitly transferring the historical interaction information to the user-specified category domain. Extensive experiments on three real-world datasets are conducted to show our IAGNN outperforms the state-of-the-art baselines in the new task.

As to future work, we prefer to involve other types of attribute information besides category information which would extend the task to an attribute aware session-based recommendation task.

## References

1. Abel, F., Herder, E., Houben, G.J., Henze, N., Krause, D.: Cross-system user modeling and personalization on the social web. *User Modeling and User-Adapted Interaction* **23**(2), 169–209 (2013)
2. Albadvi, A., Shahbazi, M.: A hybrid recommendation technique based on product category attributes. *Expert Systems with Applications* **36**(9), 11480–11488 (2009)
3. Cai, R., Wu, J., San, A., Wang, C., Wang, H.: Category-aware collaborative sequential recommendation. In: *SIGIR*. pp. 388–397 (2021)
4. Chen, T., Wong, R.C.W.: Handling information loss of graph neural networks for session-based recommendation. In: *SIGKDD*. pp. 1172–1180 (2020)
5. Choi, S.M., Han, Y.S.: A content recommendation system based on category correlations. In: *2010 Fifth International Multi-conference on Computing in the Global Information Technology*. pp. 66–70. IEEE (2010)

6. Guo, L., Tang, L., Chen, T., Zhu, L., Nguyen, Q.V.H., Yin, H.: Da-gcn: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. arXiv preprint arXiv:2105.03300 (2021)
7. He, J., Li, X., Liao, L.: Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking. In: IJCAI. vol. 17, pp. 1837–1843 (2017)
8. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
9. Hu, G., Zhang, Y., Yang, Q.: Conet: Collaborative cross networks for cross-domain recommendation. In: CIKM. pp. 667–676 (2018)
10. Kang, S., Hwang, J., Lee, D., Yu, H.: Semi-supervised learning for cross-domain recommendation to cold-start users. In: CIKM. pp. 1563–1572 (2019)
11. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
12. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. pp. 1419–1428 (2017)
13. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493 (2015)
14. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: SIGKDD. pp. 1831–1839 (2018)
15. Ma, M., Ren, P., Lin, Y., Chen, Z., Ma, J., Rijke, M.d.:  $\pi$ -net: A parallel information-sharing network for shared-account cross-domain sequential recommendations. In: SIGIR. pp. 685–694 (2019)
16. Meng, W., Yang, D., Xiao, Y.: Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In: SIGIR. pp. 1091–1100 (2020)
17. Pan, W., Xiang, E., Liu, N., Yang, Q.: Transfer learning in collaborative filtering for sparsity reduction. In: AAAI. vol. 24 (2010)
18. Pan, Z., Cai, F., Chen, W., Chen, H., de Rijke, M.: Star graph neural networks for session-based recommendation. In: CIKM. pp. 1195–1204 (2020)
19. Pang, Y., Wu, L., Shen, Q., Zhang, Y., Wei, Z., Xu, F., Chang, E., Long, B.: Heterogeneous global graph neural networks for personalized session-based recommendation. arXiv preprint arXiv:2107.03813 (2021)
20. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: CIKM. pp. 579–588 (2019)
21. Ren, P., Chen, Z., Li, J., Ren, Z., Ma, J., De Rijke, M.: Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 4806–4813 (2019)
22. Shen, Q., Wu, L., Pang, Y., Zhang, Y., Wei, Z., Xu, F., Long, B.: Multi-behavior graph contextual aware network for session-based recommendation. arXiv preprint arXiv:2109.11903 (2021)
23. Song, J., Shen, H., Ou, Z., Zhang, J., Xiao, T., Liang, S.: Islf: Interest shift and latent factors combination model for session-based recommendation. In: IJCAI. pp. 5765–5771 (2019)
24. Wang, Z., Wei, W., Cong, G., Li, X.L., Mao, X.L., Qiu, M.: Global context enhanced graph neural networks for session-based recommendation. In: SIGIR. pp. 169–178 (2020)
25. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 346–353 (2019)



26. Xu, C., Zhao, P., Liu, Y., Sheng, V.S., Xu, J., Zhuang, F., Fang, J., Zhou, X.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI. vol. 19, pp. 3940–3946 (2019)
27. Zhou, K., Wang, H., Zhao, W.X., Zhu, Y., Wang, S., Zhang, F., Wang, Z., Wen, J.R.:  $s^3$  - rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: CIKM (2020)