DAGNN: Demand-aware Graph Neural Networks for Session-based Recommendation

Liqi Yang\textsuperscript{1}, Linhao Luo\textsuperscript{1}, Fengxin Li\textsuperscript{2}, Xiaofeng Zhang\textsuperscript{1}, and Xinni Zhang\textsuperscript{1}

\textsuperscript{1} Harbin Institute of Technology, Shenzhen, China
\textsuperscript{2} PING AN INSURANCE (GROUP) COMPANY OF CHINA
liqiyanglqy@sina.com luolinhao@stu.hit.edu.cn
LIFENGXIN682@pingan.com.cn zhangxiaofeng@hit.edu.cn
zhangxinni.hit@gmail.com

Abstract. Session-based recommendations have been widely adopted for various online video and E-commerce Websites. Most existing approaches are intuitively proposed to discover underlying interests or preferences out of the anonymous session data. This apparently ignores the fact these sequential behaviors usually reflect session user’s potential demand, i.e., a semantic level factor, and therefore how to estimate underlying demands from a session is challenging. To address aforementioned issue, this paper proposes a demand-aware graph neural networks (DAGNN). Particularly, a demand modeling component is designed to first extract session demand and the underlying multiple demands of each session is estimated using the global demand matrix. Then, the demand-aware graph neural network is designed to extract session demand graph to learn the demand-aware item embeddings for the later recommendations. The mutual information loss is further designed to enhance the quality of the learnt embeddings. Extensive experiments are evaluated on several real-world datasets and the proposed model achieves the SOTA model performance.

Keywords: Recommendation System · Session-based recommendation · Graph Neural Networks.

1 Introduction

Session-based recommendation has been widely applied in various online video sites and E-commerce applications, and thus attracts a vast amount of research efforts. The prominent characteristics of session recommendation issue is that the length of an anonymous session is quite short which poses a great challenge to the conventional recommendation techniques.

Without loss of generality, most session-based recommendations are technically designed to first explore users’ preferences hidden in each session, and then best match the feature representations between target item and the extracted preferences\cite{5,10,27}. In the literature, there exist a good number of related works to extract session preferences, such as attention-based method \cite{15}. STAMP\cite{17} is proposed to investigate how the preference of the last item affects the recommendation performance. Alternatively, chained-based methods are proposed to explicitly capture the order dependencies between session items for recommendation, e.g. FPMC \cite{21} and GRU4Rec \cite{10}.
Constrained by the network structure, these chain-based approaches fail to capture the transitions among long distant items [31]. Consequently, various graph neural network models [16, 28] are proposed to capture both static or dynamic relationships, e.g., neighbors’ message passing, between distant items [20, 31, 32, 34]. Alternative to preference or interest, the intention or purpose based approaches have been proposed [29] to recommend a particular category of items to a user.

However, these existing approaches apparently ignore the fact that either intention/purpose or preference/interest is essentially driven by user’s demand, i.e., the behind reason why the user browses or buys a set of items with different categories. This paper is thus motivated to explicitly model the hidden demand for recommending next item. An illustrating example is plotted in Figure 1 to clarify the differences between different approaches. From this figure, it is clear that the intention/purpose refers to the particular category of items user want to buy. That is, a user clearly knows what kind of items to buy, e.g. a sofa. With this intention, users may have their own preferences like “light-colored or dark-colored” and “cozy or functional”. However, users may want to renovate their apartments, and thus they may buy sofa as well as other categories of items. These categories are combined together to represent either a determinant or an uncertain demand. Obviously, semantically correlated multiple intentions/purposes should be used to model such demand. Furthermore, the order dependencies among items under a demand is a prominent factor for the recommendations. For example, under the “renovation” demand, users are prone to buy “light-colored” table after choosing “light-colored” floors.

To address aforementioned issues, we propose this demand-aware graph neural network model. Particularly, we design a demand modeling component to extract session
demands as well as the demand score weighting the relevance between item category and session demand. To this end, we project the item category onto the assumed latent demand space through a linear transformation matrix, which acts as a global controller to learn the semantic correlations among item category space. Second, a demand-aware graph is constructed on the basis of items’ demand score to learn the demand-aware item embeddings. And based on the characteristics of neighbors’ message passing, we treat items’ demand score as the gate control unit to manage the messages passing between items. A mutual loss is designed to further enhance the quality of the learnt item embeddings used for the later predictions.

The major contribution of this paper are summarized as follows.

– To the best of our knowledge, we are the first to cast the session recommendation problem to the demand modeling issue. As the ground truth demand is hard to define, we carefully design the demand modeling component to extract the session demand over the whole item category space to capture semantically correlated categories. Then, we globally update the assumed demand space using each extracted session demand to approximate the underlying true demand.
– We delicately design a demand-aware graph neural network model to alleviate the loss of multi-hop neighbors’ message passing by considering the relevance between session items and its session demand. Furthermore, a mutual information loss term is designed to align the relationship between “local” item embeddings and “global” session embeddings with the relationship between item’s category and the extracted session demands.
– We perform extensive experiments on two real-world datasets and the promising results have demonstrated that the proposed approach is superior to both baseline and the state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 reviews related work and then we formulate the problem in Section 3. The proposed approach is detailed in Section 4. Experimental results are reported in Section 5 and we conclude the paper in Section 6.

2 Related Work

Session-based Recommendation Methods The purpose of session-based recommendation is to predict users’ next action based on their previous behaviors. In [23], a Markov Decision Process (MDP) is proposed to model the transition probabilities between items. Furthermore, to model the sequence relationship between two adjacent actions, [21] proposed a hybrid model, called FPMC, to combine the Matrix Factorization (MF) with Markov Chain (MC) for recommending next item. However, the MC-based models can only predict the next behavior based on user’s latest behaviors.

To address this issue, GRU based approach [10] is proposed for session-based recommendations and its paralleled version is proposed to resolve the scale issue [11]. To resolve the drift issue, [25] proposes to augment data to address the drift issue contained in the input data distribution. The milestone NARM [10] is proposed to capture user’s main purpose hidden in e session by integrating an attention mechanism with a
RNN model. Similarly, STAMP [17] is proposed to employ the basic MLP networks as well as an attentive net to capture users’ short-term interests and their general interests. Recently, various graph representation learning [7,16,32,31,30,1,20] based approaches have been proposed to learn feature embeddings in an unsupervised manner such as DeepWalk [19], LINE [26] and node2vec [8]. Then, these embeddings are directly used for the later recommendation tasks.

**Intent-aware Session-based Recommendation Methods** Several recent approaches other than extracting users’ preferences or interests have been proposed [4,18]. Similar to our approach, [29] proposed that a session should contain multiple purposes of a user, and the sequence order between items in a session should well reflect the specific purpose. Therefore, the author proposed the multi-channel purpose routers to extract user’s purposes and they also employ a recurrent model to capture item’s dependencies. However, the modeled purpose by this work is still the category of an item whereas our proposed demand is to model the semantically correlated categories. In [9], the authors considered that user’s flexible intentions are time-dependent, and their temporal characteristics are naturally modeled by the adopted LSTM component. However, the modeled user intention focuses more on capturing the mapping relationship between time steps and the items. For multi-behavior recommendation scenario, [3] directly models a sequence of category-wise user intention instead of item-wise preferences. The model needs extra information about user’s behaviors and thus does not fit for our problem. The most related work to ours is [35]. In this work, the authors model the session recommendation issue as a a hierarchical decision-making process, i.e., user first chooses an intention, and then clicks items conditionally relied on the previous clicked item. To some extent, this model is quite close to ours if there only exists one intention. However for a session containing multiple intentions, this work might not be able to successfully predict the next intention if its embeddings are far from previous ones. However, our approach may also work well for this scenario if the two intentions are semantically correlated.

3 **Problem Formulation**

Let $V = \{v_1, v_2, ..., v_s\}$ denote a sequence of $|s|$ items, the corresponding category sequence is denoted as $C = \{c_1, c_2, ..., c_s\}$. Assume that there have $M$ demand spaces and each session might contain one or several underlying demands falling into this space. Let $V, C$ respectively denote the item set and its category set, $S$ represent the session set. Given a session $s = \{(v_1, c_1), (v_2, c_2), ..., (v_s, c_s)\} | v_i \in V, c_i \in C, i \in \{1, ..., |s|\}$, our task is to predict the probability $P(v_t|s)$ that the next item $v_t$ will be recommended with respect to a session $s$.

4 **The Proposed Approach**

The proposed demand-aware graph neural network model is detailed in this section and it consists of three components, i.e., (1) demand modeling component; (2) demand-
aware item embedding component; and (3) demand-driven recommendation component. The overall framework of the proposed demand-aware recommendation model is depicted in Figure 2.

![Figure 2](image2.png)

**Fig. 2.** The overall framework of demand-aware graph neural network model. Given a session, a demand weight vector is obtained by demand modeling component. And then demand-aware item embedding is learned via gnn layer based on demand graphs constructed. At last, a session embedding with last item embedding is generated for recommendation.

![Figure 3](image3.png)

**Fig. 3.** Demand modeling component.

### 4.1 Demand Modeling Component

This component is to model the possibly contained underlying demands out of each input session.
Given a item session $V$ and its category sequence $C$, we first project $C$ to $M$ demand spaces respectively, where $M$ is a predefined value, which is formulated as

$$D^m = W_d^m C, m \in \{1, \ldots, M\},$$  

(1)

where $C \in R^{n_c \times |s|}$ denotes the category embedding sequence, $D^m \in R^{n_d \times |s|}$ denotes the representations of the $m$-th demand, and $W_d^m \in R^{n_d \times n_c}$ is the learnable weight matrix, as plotted in the upper part of Figure 3. Hence, the correlations between categories can be captured in the demand spaces.

To acquire the representations of multiple demands contained in a session, the generated demand representation is aggregated along the category direction into a demand representation vector, written as

$$d^m = \log \sum_{i=1}^{s} \exp(d^m_i),$$  

(2)

where the $i$-th column $d^m_i$ of $D^m$ represents the demand representation of the $i$-th item category in the $m$-th demand space, $d^m$ denotes the $m$-th demand representation of a current session.

After acquiring the demand representations using Eq. 1 and 2, we respectively calculate the demand score $z^m$ of session $s$ and the demand score $z^m_t$ of a target item $v_t$. To this end, we need to respectively calculate the category query vector for $s$ and $v_t$, as plotted at the lower part in Figure 3. To calculate the category query matrix $K$ and its corresponding demand score $z^m$ for session $s$, we have

$$K = W_k C$$

$$z^m = \sigma\left(\frac{(d^m)^T K}{\sqrt{n_d}}\right),$$

(4)

where $K \in R^{n_d \times |s|}$ denotes the query matrix in the $m$-th demand space, $\sigma$ is sigmoid function to normalize the score, the $i$-th $z^m_i$ of $z^m \in R^{|s|}$ denotes the contribution of the $i$-th category to the $m$-th demand $d^m$.

Similarly, to calculate the category query matrix $K_t$ and its corresponding demand score $z^m_t$ for the target item $v_t$, we have

$$K_t = W_k c_t$$

$$z^m_t = \sigma\left(\frac{(d^m)^T K_t}{\sqrt{n_d}}\right),$$

(6)

where $z^m_t$ is a real number and $z_t = [z^1_t, z^2_t, \ldots, z^M_t]^T \in R^M$.

4.2 Demand-aware Item Embedding Component

This component is proposed to learn demand-aware item embeddings. We first illustrate how the demand graphs are constructed to preserve sequential dependencies under each demand context. Then, we detail how the demand-aware item representations are learnt.
Constructing demand graphs to preserve sequential dependencies. Given a session \( s = \{(v_1, c_1), (v_2, c_2), \ldots, (v_s, c_s)\} \), we construct each directed demand graph \( G^m = (V, E^m) \) to preserve the sequence order among items, where \( V \) denotes all distinct items contained in \( s \), \( E^m \) denotes the edge set of \( G^m \) under the \( m \)-th demand space.

For a node \( v_i \in V \) in the graph \( G^m \), its demand score \( z_i^m \) is already acquired from the previous component. To generate edges, we first construct the directed solid edges according to the order that items appear in a session. That is, there will be a solid edge \( e_{ij} = 1 \) between item \( v_i \) and item \( v_j \) if \( j = i + 1 \). To further consider the effect of the demand, message passing between items in a demand graph might be different. For instance, in the left figure in Figure 4, items \( v_5, v_1, v_3, v_4 \) belongs to the same demand, but it is impossible for item \( v_5 \) to obtain the information from its 2-hop neighbors \( v_1, v_4 \) under the same demand. The reason is that the one-hop neighbor node \( v_2 \) does not belong to the same demand as node \( v_5 \) does, which results in a lower \( z_2^m \) score. Therefore, a dashed edge is inserted between item \( v_i \) and \( v_j \) if \( 0 < j - i < k + 1 \) and \( k \) is a predefined size of sliding window. Furthermore, in order to obtain the

\[
e_{ij}^m = \log(z_i^m z_j^m + 1)
\]

Demand-aware item representation learning. After constructing demand graphs, we learn the item embeddings in this subsection. The general aggregation function, such as mean, sum, and LSTM aggregators cannot be directly applied to well consider the demand information.

Inspired by the gate control mechanism [13,24] and the graph attention neural network [28], we treat the demand score of a target node as the gated unit, which is a soft gate to control messaging passing from neighboring nodes to the target node. Hence, for each node \( v_i \) in graph \( G^m = (V, E^m) \), the aggregation of neighboring nodes in the

Fig. 4. The left graph is constructed only according to items’ order in a session, and the right graph adds additional dashed edges.
$l$-th layer of GNN is directly defined as

$$h_{m,l}^i = \text{agg}(\{v_{j}^{m,l-1}, j \in N_{v_i}\})$$

(8)

$$\text{agg}(\cdot) = \frac{1}{|N_{v_i}|} \sum_{j \in N_{v_i}} z_{j}^m e_{ij} v_{j}^{m,l-1}$$

(9)

$$\hat{h}_{m,l}^i = \text{agg}(\{v_{j}^{m,l-1}, j \in \hat{N}_{v_i}\})$$

(10)

$$\text{agg}(\cdot) = \frac{1}{|\hat{N}_{v_i}|} \sum_{j \in \hat{N}_{v_i}} z_{j}^m e_{ij} v_{j}^{m,l-1},$$

(11)

where $N_{v_i}$ denotes neighboring nodes of $v_i$ connected by solid edges, and $\hat{N}_{v_i}$ denotes neighboring nodes connected by dashed edges. Now, the item embedding of $v_i^m$ is updated using

$$v_{i}^{m,l} = \sigma(W_{m,l}^{m,l} \text{concat}(v_{i}^{m,l-1}, h_{m,l}^{i}, \hat{h}_{m,l}^{i}) + b_{m}),$$

(12)

where $W_{m,l}^{m,l} \in \mathbb{R}^{n \times 3n}$ and $b_{m} \in \mathbb{R}^{n}$ is the bias.

4.3 Demand-driven Recommendation Component

The proposed component is plotted at the right hand side of the Figure 2. In this component, we first embed each session and then predict the probability whether the target item will be recommended or not.

Session Embedding With the learnt demand-aware item embeddings, we further learn the demand-aware session representations $s_{g}^{m}$ through a readout function of graph $G_{m} = (V, E_{m})$. By following [31], a soft-attention mechanism is applied to aggregate all the node embeddings, calculated as

$$\beta_{i}^{m} = q^{T} \sigma(W_{1} v_{i}^{m} + W_{2} v_{i}^{last} + b_{1})$$

(13)

$$s_{g}^{m} = \sum_{i=1}^{|N_{s}|} \beta_{i}^{m} v_{i}^{m},$$

(14)

where $W_{1}, W_{2} \in \mathbb{R}^{n \times n}$, $q$ and $b_{1} \in \mathbb{R}^{n}$ are learnable parameters, $|N_{s}|$ is the session length.

Having item embedding and session embedding, it is desired to align the relationship between “local” item embeddings and “global” session embeddings with the relationship between item’s category and the extracted session demands. Inspired by contrastive learning [2,6,12], we carefully design a mutual information loss to minimize the difference between global representation $s_{g}^{m}$ and local item representation, calculated as

$$L_{MIM} = \frac{1}{|N_{s}|M} \sum_{m=1}^{M} \sum_{i=1}^{|N_{s}|} E_{(V,E_{m})} \{ z_{i}^{m} \log D(s_{g}^{m}, v_{i}^{m}) \}
+ (1 - z_{i}^{m}) \log(1 - D(s_{g}^{m}, v_{i}^{m})) \},$$

(15)
where the discriminator $D(\cdot)$ is a bilinear function to measure whether an item belongs to the current session or not w.r.t. a specific demand, $z^m_i$ denotes the demand score of item $v_i$ in the $m$-th demand space.

**Prediction layer** To prediction whether a target item would be recommended or not, we first calculate the session embeddings $s^m_{mg}$ and the target item embeddings $v^m_{last}$ under the $m$-th demand. Then, a linear transformation is applied to concatenate these embeddings together to form the input $s^m$ of the prediction layer, written as

$$s^m = W^h \text{concat}(s^m_{mg}, v^m_{last}),$$  \hspace{1cm} (16)

where $W^h \in \mathbb{R}^{n \times 2n}$ is the network parameters. The probability $P(v_t|s)$ to recommend a target item $v_t$ is now given as

$$P(v_t|s) = \text{Softmax}(\sum_{m=1}^{M} z^m_t s^m T v_t),$$  \hspace{1cm} (17)

where $z^m_t$ denotes the demand score of item $v_t$ in the $m$-th demand space.

The cross entropy loss is adopted as the model loss, computed as

$$\mathcal{L}_p = \sum_{s \in S} y_{v_t} \log P(v_t|s) + (1 - y_{v_t}) \log(1 - P(v_t|s)),$$  \hspace{1cm} (18)

where $S$ denotes a set of sessions, $y_{v_t}$ is the ground truth of the target item. Accordingly, the overall model loss is calculated as

$$\mathcal{L} = \mathcal{L}_p + \lambda_1 \mathcal{L}_{MIM} + \lambda_2 ||\Theta||_2,$$  \hspace{1cm} (19)

where $\lambda_1$ and $\lambda_2$ are hyper-parameters to balance the effect of mutual information loss and the $L_2$ regularizations, and $\Theta$ is the set of learnable network parameters.

## 5 Experimental Results

In this section, we first briefly introduce experimental datasets, evaluation metrics as well as the experimental settings. Then, to evaluate the model performance, both the baseline and the state-of-the-art session based recommendation approaches are compared to evaluate the effectiveness of our approach. Extensive experiments are then evaluated on two real-world datasets to answer following research questions:

- **RQ1**: Whether the proposed approach outperforms the state-of-the-art approaches for session recommendation tasks or not?
- **RQ2**: Does the category information affect the model performance or not?
- **RQ3**: Whether the proposed components, i.e., demand modeling component, demand-aware graph neural network component and the mutual information loss, could affect the model performance or not (ablation study)
- **RQ4**: How the model parameters affect the model performance (Parameter Analysis).
- **RQ5**: A case study on how the proposed demand could affect the model performance.
5.1 Experimental Setting

Datasets and Evaluation Criteria In the experiments, two widely adopted real-world datasets are adopted to evaluate the model performance which are respectively reviewed as follows.

- **Tmall dataset** is first released by the competition held by IJCAI-15. It consists of purchase records collected from users’ transactions on Tmall platform, and each record only contain users’ purchase behavior information. To pre-process the data, by following [33], items purchased by a user within one day are treated as a session, and we filter out those sessions if their length is less than 5, and we further remove the items if their session frequency is less than 30.

- **Tafeng dataset** is released by the Kaggle competition and it contains transaction data collected from a Chinese grocery store. To pre-process the data, items purchased by a user within one day are treated as a session, and we delete those sessions whose length is less than 2. Then, we filter out items whose session frequency is less than 30.

In addition, we select the first 80% of the data as the training set, and the remaining 20% as the test set according to its chronological order. The statistics of these two datasets are reported in Table 1. Two widely adopted evaluation criteria, i.e., Recall and NDCG, are chosen to evaluate the model performance.

| Dataset | #item  | #category | #avg.session length | #train session | #test session |
|---------|--------|-----------|---------------------|----------------|---------------|
| Tmall   | 17,095 | 698       | 6.175               | 28,004         | 7,001         |
| Tafeng  | 14,637 | 1,638     | 7.814               | 80,552         | 20,138        |

Baselines In order to evaluate the model performance of the proposed approach, both the baseline models as well as the state-of-the-art approaches are compared in the experiments, and we briefly review them as follows.

- **POP** recommends items based on the item popularity. Although it is considered as the baseline model, it usually achieves a superior model performance on various datasets.

- **item-KNN** [22] is considered as the baseline model although it usually achieves the SOTA performance on many recommendation tasks, and it recommends items that are most similar to the clicked items in current session.

- **FPMC** [21] a hybrid model which is proposed to integrate the first-order Markov Chain with the matrix decomposition. To adapt this method for session based recommendations, we simply ignore users’ latent representations and treat all the items in a session as a basket of items to recommend the next item.

- **GRU4Rec** [10] is proposed to employ the Gated Recurrent Unit (GRU) to capture order dependencies between items for representing users’ preferences.
– **NARM** [15] is one of the milestone recommendation systems. It integrates an attention mechanism with RNNs to model users’ sequential behaviors.

– **SR-GNN** [31] is considered as the SOTA model. It introduces a gated GNN layer to a directed session graph, and then obtains item embeddings. A soft-attention mechanism is adopted to aggregate items representations as the global preference, which is concatenated with the embedding of the last item to recommend the next item.

**Parameter Setting** For all the compared methods, we keep the same parameter setting as their original work does. For our method, all model parameters are initialized using a uniform distribution, then the Adam optimizer [14] is adopted to optimize the model with the initial learning rate is set to 0.001, and the L2 penalty is set to 10-5 to prevent overfitting. The embedding dimension is fine-tuned to 128 for Tafeng dataset and 100 for Tmall dataset.

| Methods   | R@20   | R@40   | NDCG@20 | NDCG@40 |
|-----------|--------|--------|---------|---------|
| POP       | 6.6995 | 10.5327| 3.5629  | 4.3399  |
| item-KNN  | 10.289 | 12.9556| 5.6112  | 6.1557  |
| FPMC      | 9.3058 | 12.9755| 4.5507  | 5.3119  |
| GRU4Rec   | 8.6116 | 12.2331| 4.7705  | 5.5075  |
| NARM      | 7.9936 | 11.8413| 3.9478  | 4.5909  |
| SR-GNN    | 8.3119 | 11.8695| 4.5897  | 5.3132  |
| GRU4Rec*  | 9.8007 | 13.6168| 4.5822  | 5.365   |
| NARM*     | 9.7466 | 13.9168| 4.6854  | 5.5396  |
| SR-GNN*   | 11.2129| 15.6764| 5.1335  | 6.0464  |
| **Our method** | **11.6363** | **15.9488** | **5.8516** | **6.7319** |
| Improve(%)| 3.776  | 1.738  | 4.284   | 9.360   |

Table 2. Model performance evaluation results on Tafeng dataset. Method annotated with an asterisk denotes that the model is trained with item category information.

### 5.2 RQ1 & RQ2: Performance Comparison

We evaluated all the compared methods as well as ours on two real-world datasets, and reported the corresponding results in Table 2 and Table 3, respectively. The compared approaches annotated with ‘*’ are trained using item’s category information, whereas the rest ones are trained with item IDs. From Table2,3, the following observations could be found, listed as

– Our proposed approach consistently outperforms all the baselines trained with or without category information.
Table 3. Model performance evaluation results on Tmall dataset. Method annotated with an asterisk denotes that the model is trained with item category information

- The item category information plays a crucial role in recommending the next items, and this could be seen from the results of “GRU4Rec*”, “NARM*” and “SR-GNN*” from both Tables, where their evaluation scores are higher than their counterpart models. Although the NDCG results of “GRU4Rec*” slightly decreases on Tafeng dataset, its recall results obviously increase.
- As the SOTA approach, the “SR-GNN*” achieves the best recall scores on both Tmall and Tafeng datasets, the best NDCG scores on Tmall and second best NDCG scores on Tafeng. This partially verifies that the effectiveness of graph neural network-based approaches.

5.3 RQ3: Ablation Study

In this experiment, we respectively remove the proposed demand modeling (DM), demand aware GNN (DGNN) and mutual information loss (MI), and the experimental results are reported in Table4.

Obviously, after removing demand-aware GNN component, the model performance dramatically decreases w.r.t Recall and NDCG metrics. This verifies that demand-aware GNN could well learn item embeddings at the demand level. If removing demand modeling component, the model performance decreases on Tafeng dataset, but is not significant for Tmall dataset, e.g., its NDCG value slightly decreases. Furthermore, if removing the mutual information loss, the model performance significantly decreases on Tafeng dataset but is not the case for Tmall dataset. This demonstrates that the necessity of aligning the relationship between “local” item embeddings and “global” session embeddings with the relationship between item’s category and the extracted session demands, for the scenario where the underlying demands dominate the later recommendations.
Table 4. Results of ablation studies on Tafeng and Tmall.

| Methods   | Tafeng       | Tmall       |
|-----------|--------------|-------------|
|           | R@20 | R@40 | NDCG@20 | NDCG@40 | R@20 | R@40 | NDCG@20 | NDCG@40 |
| DAGNN-DM  | 10.65 | 14.98 | 3.01 | 5.89 | 26.30 | 30.80 | 16.68 | 17.59 |
| DAGNN-GNN | 9.37 | 12.92 | 4.59 | 5.31 | 24.13 | 28.08 | 15.50 | 16.31 |
| DAGNN-MI  | 10.53 | 14.99 | 4.63 | 5.54 | 26.19 | 30.53 | 17.04 | 17.92 |
| DAGNN     | 11.63 | 15.95 | 5.85 | 6.73 | 26.33 | 30.59 | 17.21 | 18.09 |

5.4 RQ4: Parameter Analysis

This subsection evaluates the effect of model parameters which are respectively reported as follows.

Effect of the number of demands We vary the number of demands from 1 to 4 and evaluate the model performance on two datasets, and the results are reported in Figure 5.

It is noticed that the model performance is the best when demand number = 2 on Tafeng, whereas the model performance is the best when demand number = 3 on Tmall dataset. For Tmall dataset, the effect of the demand number is not obvious compared to that of the Tafeng dataset. The essential reason behind this is that Tmall is not a ‘demand-driven’ dataset, the category in Tmall dataset is relative small and the underlying demand is also implicit. This is also why the demand modeling poses small impact to the model performance for Tmall dataset in Table 4.

![Fig. 5. Effect of demand number on model performance.](a) Tafeng Dataset (b) Tmall Dataset)

Effect of the number of GNN layers We vary the number of GNN layers from 1 to 4 and evaluate the model performance on two datasets, and the results are reported in Figure 6. We noticed that the model performance gradually decreases with the number
of GNN layers increases. This indicates that a simple network structure is preferred if the structure of input data is relative simple, and this also explains why the conventional item-KNN could achieve the superior performance.

![Fig. 6. Effect of the number of GNN layers on model performance.](image)

### 5.5 RQ5: A case study of demand

To assess the effect of the proposed demand, we perform this case study. In this study, we choose two sessions, with each has two underlying demands, from Tafeng and Tmall dataset, respectively, and the demand score of each item is reported in Table 5. From this table, it is clear that for Tafeng data, the 1-th demand is more important than the 2-th demand for the first item, whereas the 2-th demand become more important than the 1-th demand for the last item. Similar observations could be found from Tmall data. This partially verifies our assumption that the session based recommendation should be carefully designed for discovering the multiple underlying demands and then differentiate their importance for the recommendations.

| Tafeng | item id | 1087 | 8544 | 7238 | 6422 | 2355 |
|--------|---------|------|------|------|------|------|
| 1-th demand score | 2.87 | 1.49 | 1.53 | 0.59 | 0.78 |
| 2-th demand score | 1.28 | 2.46 | 2.13 | 2.13 | 2.86 |

| Tmall | item id | 10807 | 11908 | 2801 | 1645 | 5492 |
|-------|---------|-------|-------|------|------|------|
| 1-th demand score | 3.42 | 2.6 | 2.24 | 2.91 | 1.44 |
| 2-th demand score | 2.15 | 2.72 | 2.74 | 2.48 | 2.82 |

| Table 5. The case study results on demand effect. |
6 Conclusion

In this paper, we propose a novel demand-aware graph neural networks for session-based recommendation. Particularly, we design a demand modeling component to extract session demand which is used to guide the construction of the weighted session graphs for the learning of demand-aware item embeddings. A mutual information is further designed to align item’s embedding within each demand. Extensive experiments are evaluated on two real-world datasets and the proposed model achieves the superior model performance over both the baseline models and the SOTA approaches.
References

1. Chen, T., Wong, R.C.W.: 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. pp. 1172–1180 (2020)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
3. Chen, T., Yin, H., Chen, H., Yan, R., Nguyen, Q.V.H., Li, X.: Air: Attentional intention-aware recommender systems. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). pp. 304–315. IEEE (2019)
4. Chen, W., Ren, P., Cai, F., Sun, F., de Rijke, M.: Improving end-to-end sequential recommendations with intent-aware diversification. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 175–184 (2020)
5. Feng, Y., Lv, F., Shen, W., Wang, M., Sun, F., Zhu, Y., Yang, K.: Deep session interest network for click-through rate prediction. arXiv preprint arXiv:1905.06482 (2019)
6. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. pp. 249–256. JMLR Workshop and Conference Proceedings (2010)
7. Gori, M., Monfardini, G., Scarselli, F.: A new model for learning in graph domains. In: Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005. vol. 2, pp. 729–734. IEEE (2005)
8. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
9. Guo, X., Shi, C., Liu, C.: Intention modeling from ordered and unordered facets for sequential recommendation. In: Proceedings of The Web Conference 2020, pp. 1127–1137 (2020)
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
11. Hidasi, B., Quadrana, M., Karatzoglou, A., Tikk, D.: Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM conference on recommender systems. pp. 241–248 (2016)
12. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:1808.06670 (2018)
13. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015)
15. 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)
16. Li, Y., Tarlow, D., Brockschmidt, M., Zemel, R.: Gated graph sequence neural networks. arXiv preprint arXiv:1511.05493 (2015)
17. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: 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. pp. 1831–1839 (2018)
18. Liu, Z., Chen, H., Sun, F., Xie, X., Gao, J., Ding, B., Shen, Y.: Intent preference decoupling for user representation on online recommender system. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (2020)
19. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
20. Qiu, R., Li, J., Huang, Z., Yin, H.: 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. pp. 579–588 (2019)
21. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th international conference on World wide web. pp. 811–820 (2010)
22. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295 (2001)
23. Shani, G., Heckerman, D., Brafman, R.I., Boutilier, C.: An mdp-based recommender system. Journal of Machine Learning Research 6(9) (2005)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv preprint arXiv:1409.3215 (2014)
25. Tan, Y.K., Xue, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st workshop on deep learning for recommender systems. pp. 17–22 (2016)
26. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: Large-scale information network embedding. In: Proceedings of the 24th international conference on world wide web. pp. 1067–1077 (2015)
27. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. pp. 565–573 (2018)
28. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
29. Wang, S., Hu, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Cao, L.: Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In: IJCAI. vol. 19, pp. 3771–3777 (2019)
30. 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: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 169–178 (2020)
31. 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)
32. 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)
33. Ying, H., Zhuang, F., Zhang, F., Liu, Y., Xie, X., Xiong, H., Wu, J.: Sequential recommender system based on hierarchical attention network. In: IJCAI International Joint Conference on Artificial Intelligence (2018)
34. Zhang, M., Wu, S., Gao, M., Jiang, X., Xu, K., Wang, L.: Personalized graph neural networks with attention mechanism for session-aware recommendation. IEEE Transactions on Knowledge and Data Engineering (2020)
35. Zhu, N., Cao, J., Liu, Y., Yang, Y., Ying, H., Xiong, H.: Sequential modeling of hierarchical user intention and preference for next-item recommendation. In: Proceedings of the 13th International Conference on Web Search and Data Mining. pp. 807–815 (2020)