Knowledge-enhanced Multi-View Graph Neural Networks for Session-based Recommendation

Anonymous Author(s)
Submission Id: 6378

ABSTRACT

Based on the short-term behavior of anonymous users, session-based recommendation (SBR) has gained increasing attention to predict the next item via extracting and integrating intra-session item features. Limited by the sparsity problem caused by short-term behavior, global item correlations across sessions have begun to be mined and exploited to enhance session representations. However, most SBR methods only rely on chronological co-occurrence relationships to model global item correlation and neglect the ground-truth semantic correlations, resulting in redundant mining of sequential correlations and weak accuracy performance. Inspired by the knowledge graph (KG) that can accurately express the real semantic relationship between items, we propose a novel Knowledge-enhanced Multi-View Graph Neural Network (KMVG) to improve session-based recommendation by utilizing multiple relationships between items. Specifically, we first exploit the graph neural network(GNN) to learn item representations from the session view (i.e., short-term behavior sequence) and the knowledge view (constructed by knowledge graph), respectively. Besides, the pairwise view is built and exploited to explore the feature commonality between items within a session for enhancing session representations in a fine-grained manner. By combining the session representations from multiple views, KMVG can make full use of multiple item relationships (global and local) to achieve high-quality next item prediction. Extensive experiments on three real-world public datasets demonstrate that KMVG outperforms the state-of-the-art baselines. Further analysis also reveals the effectiveness of KMVG in exploiting the item-item relationships under multiple views.

CCS CONCEPTS

• **Information systems** → *Recommender systems.*

KEYWORDS

Session-based Recommendation, Graph Neural Network, Knowledge Graph

ACM Reference Format:

Anonymous Author(s). 2018. Knowledge-enhanced Multi-View Graph Neural Networks for Session-based Recommendation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 9 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Recommender system (RS) is an effective mechanism to help users find items they are interested in from a large number of items to resist the problem of *information overload*. As the most popular thought, collaborative filtering (CF) usually relies on user profiles and the long-term historical interactions between users and items [9, 16]. However, in some real-world scenarios, where user information is unavailable (e.g., the unlogged-in user or anonymous user) or the interaction information is limited (e.g., short-term historical interaction produced by a new user or an inactive user), conventional CF approaches may perform poorly. Consequently, session-based recommendation (SBR) comes into being and has attracted extensive attention recently. Each session consists of multiple user-item interactions in chronological order that occurred over a short continuous period, and the basic task of SBR is to predict the next item based on a given anonymous session.

Some early studies based on Markov chains [15, 17] predict the possible next item by modeling the transformation of items in a session. Under the strong independence assumption, independent combinations of the past user behaviors confine the prediction accuracy. Later, some researchers have spent a lot of efforts on SBR by introducing Recurrent Neural Networks (RNNs) and have achieved remarkable results [5, 10, 20, 21, 24]. However, due to the well-known problem of long-distance dependencies and only modeling one-way transitions between consecutive items, these RNN-based methods are insufficient to mine the transitional relationships between items within a session to model item representations. Based on its strong ability in mining higher-order connectivity, recent works widely utilize graph structure [1, 31, 33] to represent the session and adopt Graph Neural Networks (GNNs) to propagate information between adjacent items. The results of these works demonstrate the superiority of GNN-based methods over traditional methods. However, limited by the sparsity problem caused by short-term behavior, only mining and exploiting the relationship between items within the session to improve the representation ability and performance of the model has encountered a bottleneck.

More recent works have contributed to extracting and embedding cross-session item relations as a complement to intra-session item representation [14, 30, 34]. These studies typically exploit the co-occurrence relation of items cross sessions to build an additional global item-item graph, and then utilize graph neural networks to propagate and aggregate information about co-occurring items on this graph. Figure 1 (a) illustrates an example of global item-item graph construction. In particular, we first count the number of times every two items appear in the same session, e.g., item v_1 and item v_2 co-occur 3 times in different sessions. Then, we filter the top 3 items with the highest number of co-occurrences as neighbors of

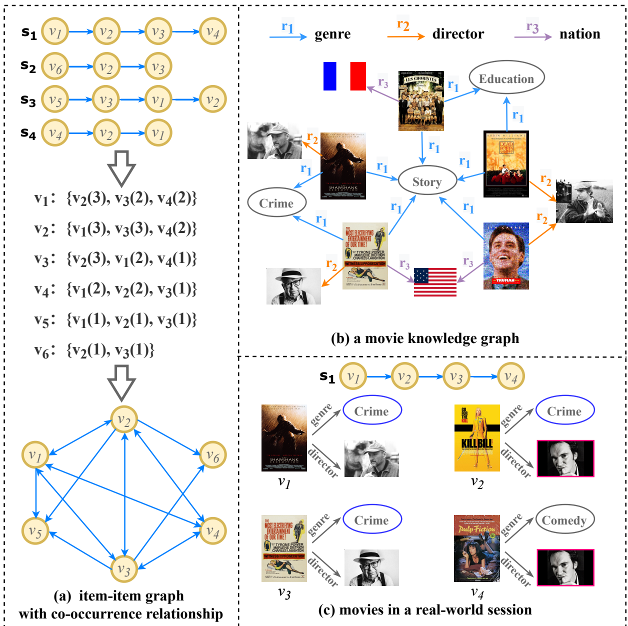


Figure 1: Illustration of item-item relationships. (a) shows the process of constructing the global item-item graph with a co-occurrence relationship. (b) is a real-world movie knowledge graph. (c) shows the two explicit attributions of movies within a session.

the current item, e.g., items v_2, v_3, and v_4 are taken as the three most relevant neighbors of item v_1. Obviously, we can observe that the neighbors of an item in Figure 1 (a) are necessarily the contextual neighbors of the item in a session, leading to a strong sequence correlation between the item and its neighbors. Using higher-order connectivity to extract item features in the session graph and global item-item graph respectively will result in redundant mining of sequence dependencies, which is limited and inappropriate for modeling item representations. Therefore, it is desirable to exploit and excavate more reasonable and real relationships between items to improve the integrated item representation quality, thereby enhancing the performance of recommendation.

For avoiding sequence-dependent redundant mining, we consider two extra sequence-independent item relationships:

- **Global item-item relationship**. We emphasize that the global item-item relationship should be relatively realistic to alleviate the sparsity problem caused by short-term behavior sequences. Knowledge graph (KG) is a good choice because it contains the real semantic relationship between items and links items via the common attributes. Moreover, KG has been proven to alleviate interaction sparsity in traditional collaborative filtering. Thus, we introduce the knowledge graph to capture the global item-item relation. Figure 1 (b) shows a toy of real-world knowledge graph about movies with three relations (i.e., *genre*, *director* and *nation*). In Figure 1 (b), semantic information of items can be transferred along relational paths. Such a global item-item relationship is sequence-agnostic and can be used to mine and exploit rich external knowledge to alleviate the session sparsity problem.
- **Local item-item relationship**. Simply aggregating all items in a session to model session representations lacks a keen sense of the commonality of item representations and may introduce item features irrelevant to user interests. For instance, in Figure 1 (c), three movies v_1, v_2, and v_3 in the same session s_1 explicitly have the same type *crime*, while movies v_2 and v_4 have the same director. We hence can infer that the user's interest in this session is mainly focused on common attribution features, i.e., *crime* and *Quentin Tarantino*, and other features may be irrelevant. Therefore, in addition to the sequence relationship in the session, mining the feature commonality between items in the local item-item relationship can further improve the representation ability of the session in a fine-grained pattern, so as to more accurately represent the user's preference in the session.

To utilize the above relationships to alleviate the issue faced by SBR with short-term behavior, we propose a novel Knowledge-enhanced Multi-View Graph Neural Network (KMVG) to enhance session representations by mining and exploiting global or local relationships among items under three views, *knowledge view*, *session view* and *pairwise view*. Specifically, for the knowledge view, we employ a knowledge graph attention network to aggregate item semantic representations with rich knowledge information via capturing global item-item semantic correlation. For the session view, based on different edge relations, we develop a graph attention network to extract item representations with contextual patterns in a session by taking the item semantic embeddings and initial embeddings as input. For the pairwise view, a simple but effective pairwise item aggregator is designed to extract feature commonality among items within a session for expressing the fine-grained user preference of the current session. Afterward, we adopt a position-aware soft-attention mechanism to fuse the item embeddings from the session view to represent the current session. Finally, the session representations from different views are further integrated for the next item prediction. The main contributions of our work are:

- We emphasize the importance of global and local item-item relationships for alleviating the session sparsity. As far as we know, this is the first work to construct global item-item relationships by introducing knowledge graph into SBR.
- We develop a novel multi-view graph neural network named KMVG for SBR, which extracts fine-grained session representation by adopting tailored graph networks to mine and utilize global or local relationships between items under three views.
- Extensive experiments on three real-world datasets demonstrate that KMVG outperforms state-of-the-art baselines, and further results also verify the effectiveness of KMVG in capturing multiple relationships among items to improve model performance.

2 RELATED WORK

We review some related works on session-based recommendations, including conventional SBR methods, GNN-based SBR methods, and some works of recommendation based on KG.

Conventional SBR Methods. Early methods based on Markov chains predict the next-click item through the previous clicks and treat the SBR as a sequential optimization problem. Some work [15, 17] employ Markov decision processes to solve the problem. However, the strong independence assumption followed by these methods confines the prediction accuracy. To model the sequential relationships between items in a session, RNNs are paid much attention by many researchers for SBR [5, 7, 10, 20]. For example, GPU4Rec uses Gated Recurrent Unit (GRU) [2] to model the sequential behavior of items in a session [5]. Li *et al.* propose NARM by employing an attention mechanism on RNNs to capture user's intent of sequential behavior [10]. However, these methods only model the single-way transitions between consecutive items, neglecting the complex contextual transitions.

GNN-based SBR Methods. Due to its strong representation learning ability, GNN has been widely used in subsequent session-based recommendation methods [30, 31, 33, 34]. SR-GNN is a typical model to capture the transition information between adjacent items by applying gated graph neural network [31]. GC-SAN uses a self-attention mechanism to learn the global dependencies between distant positions [33]. Though GNN-based methods have shown a promising direction, the session graphs face a lossy encoding problem. To solve the problem, Chen *et al.* proposed to aggregate information effectively by lossless edge-order preserving aggregation and shortcut graph attention [1]. Pan *et al.* proposed to use a star node to bridge items by filtering out irrelevant items [13]. Xia *et al.* proposed to augment session data by co-training to capture the intent of users more accurately [32]. Zhang *et al.* introduce the price factor into session-based recommendation [36] and Guo *et al.* use heterogeneous graph attention network to get the intent representations from multi-granularity levels [3]. However, limited by the session sparsity problem caused by short-term behavior, only mining and exploiting the relationship between items within the session is insufficient to improve the session representation. Recently, CSRM is proposed to incorporate the collaborative information via a memory network to enrich the representation of the current session, boosting the performance of recommendation [14]. GCE-GNN treats item as the minimum granularity to learn transition information from global-level and session-level graphs [30]. Ye *et al.* proposed CA-TCN to explore the cross-session influence on item-level and session-level simultaneously [34]. While encouraging, these works usually only utilize co-occurrence relationships between items, resulting in redundant mining of sequence correlations, which is not conducive to recommendation performance.

KG-based Recommendation. Knowledge graphs are used in modern recommender systems to alleviate the data sparsity problem due to their rich semantic information. Typically, CKE is the first work to use structured information of knowledge base for collaborative filtering by utilizing TransR [11] as transform [35]. Later, Huang *et al.* propose KSR to integrate with external memories by leveraging knowledge graph information on sequential recommendation [6]. KGNN-LS is proposed to convert KG into user-specific graphs and consider user preference on KG relations and label smoothness in the information aggregation phase to generate user-specific item representations [23]. CKAN is a further improvement of KGNN-LS to utilize different neighborhood aggregation schemes on the user-item graph and KG to obtain user and item embeddings [29]. KGAT combines KG with the user-item graph and applies an attentive neighborhood aggregation mechanism on a holistic graph [25]. KGIN further reveals user intents behind the interaction with KG information [27]. In this paper, we consider using knowledge graphs as global item-item relationships to mine ground-truth semantic information.

3 PRELIMINARIES

In this section, we first present the formal definition of the general session-based recommendation problem and then introduce the three view graph structures, i.e., *knowledge-view graph*, *session-view graph* and *pairwise-view graph*, based on different views of item transitions over sessions for learning session representations.

3.1 Problem Statement

Let $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ represents all of the items. Each anonymous session consisting of a sequence of interactions (e.g., items clicked by a user) in chronological order is denoted as $\mathcal{S} = \{v_1^s, v_2^s, ..., v_l^s\}$, where v_i^s denotes item v_i clicked within session $\mathcal{S}$, l denotes the length of session $\mathcal{S}$. Given a session $\mathcal{S}$, the problem of session-based recommendation aims to recommend the next item v_{l+1}^s from $\mathcal{V}$ that is most likely to be clicked by the user of current session $\mathcal{S}$. Therefore, we can formalize this process as,

$$p(v_{l+1}^s|\mathcal{S}) = \textsc{Predict}(\mathcal{S}, \mathcal{V}) \tag{1}$$

where $p(v_{l+1}^s|\mathcal{S})$ is the probability distributions of the predicted next item, $\textsc{Predict}(\cdot)$ is a prediction function to generate the probability of next item that may be clicked by taking the session $\mathcal{S}$ and item space $\mathcal{V}$ as input.

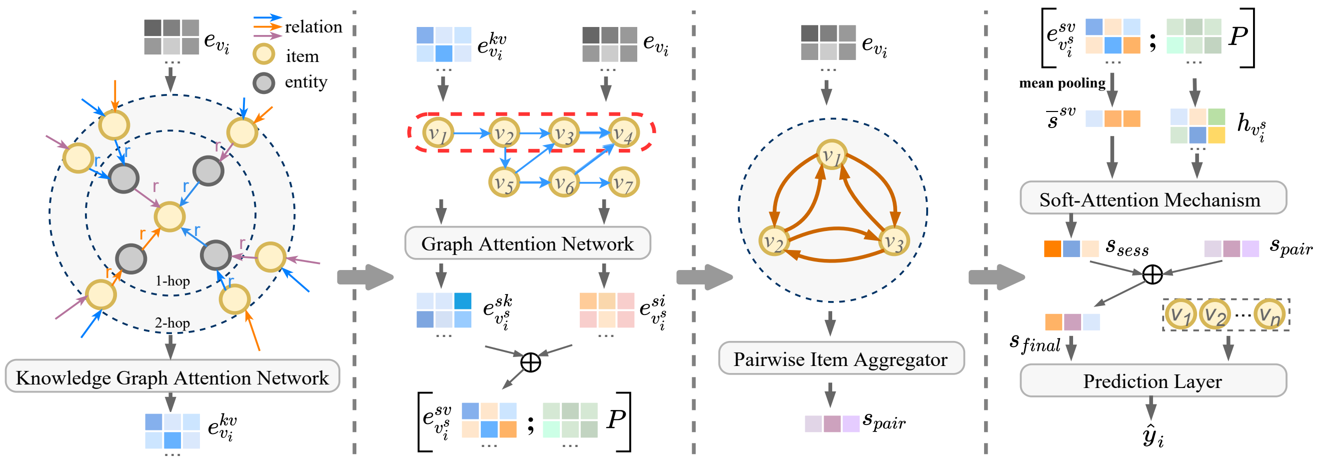
3.2 Multi-View Graph Models

As mentioned earlier, we introduce three views to capture different levels of relationship between items for modeling item and session representations. Next, we describe the graph structures and item dependency information of the three views in detail.

3.2.1 ***Knowledge-View Graph.*** We denote knowledge graph as $\mathcal{G}_{kv} = \{(h, r, t)|h, t \in \mathcal{E}_{kv}, r \in \mathcal{R}\}$, where $\mathcal{R}$ and $\mathcal{E}_{kv}$ represents the relation and the entity sets, respectively. Each triple (h, r, t) means there are a relation r from head entity h to tail entity t. Noteworthy, knowledge graph $\mathcal{G}_{kv}$ is composed of all items $\mathcal{V}$ and its attribute entities. Without loss of generality, we use h and t to perform subsequent computations uniformly.

3.2.2 ***Session-View Graph.*** We use items $\mathcal{S}$ in a sessions to construct a directed session graph, which is denoted as $\mathcal{G}_{sv} = (\mathcal{V}, \mathcal{E}_{sv})$, where $\mathcal{E}_{sv} = \{(v_i^s, v_j^s)|v_i^s, v_j^s \in \mathcal{S}, v_j \in \mathcal{A}(v_i)\}$, $\mathcal{A}(v_i)$ denotes the next click of v_i. Following [30], we define four types of edges: $r_{in}, r_{out}, r_{in-out}, r_{self}$. For edge (v_i, v_j), r_{in} represents there is only transition from v_j to v_i, r_{out} means transition only exists on v_i to v_j, r_{in-out} indicates there is a bidirectional transition between v_i and v_j, r_{self} refer to a self-loop of an item. The complex contextual transition of items can be learned under the session-view graph.

3.2.3 ***Pairwise-View Graph.*** Pairwise-view graph is constructed by connecting pairs of items in a session. In particular, we denote pairwise-view graph as $\mathcal{G}_{pv} = \{(v_i^s, v_j^s)|v_i^s, v_j^s \in \mathcal{S}\}$, where v_i^s and v_j^s are any two items in the same session.



(a) Knowledge-View (b) Session-View (c) Pairwise-View (d) Ensemble and Prediction

Figure 2: Illustration of the proposed KMVG.

4 METHODOLOGY

Figure 2 depicts the framework of our proposed KMVG, which mainly comprises four modules: 1) ***knowledge-view representation learning***, which learns knowledge-enhanced item representations by employing relation graph aggregation operation to incorporate each item's semantic neighbors' embeddings based on knowledge graph; 2) ***session-view representation learning***, which utilizes a graph convolutional network to learn session-level item representations by capturing the sequence correlation on session-view graph; 3) ***pairwise-view representation learning***, which mines and aggregates the latent feature commonality of pairwise items within the current session to represent the session embeddings; and 4) ***ensemble and prediction***, which combines the item and session representations learned from the above three views to predict the probability of candidate items for recommendation. We now introduce the four modules in detail.

4.1 Knowledge-View Representation Learning

The main task of knowledge-view representation learning is to extract the semantic information of items from the global knowledge graph. An item v_1 in general is involved in multiple triplets, e.g., the one-hop relationship $v_1 \xrightarrow{r_1} e_1$ and $v_1 \xrightarrow{r_2} e_2$, where e_1 and e_2 are the neighbor entities that have relations r_1 and r_2 with item v_1, respectively. Then, v_1 can aggregate multiple relations of neighbor entities e_1 and e_2 to enrich its own semantics. Furthermore, the semantic information can be propagated and transformed between different items through two- (or more) hop relationships, e.g., $v_1 \xrightarrow{r_1} e_3 \xrightarrow{-r_1} v_2$.

To realize the flow and transformation of such semantic information on knowledge graph, we employ a knowledge graph attention network to recursively propagate item embeddings along high-order connectivity of knowledge graph [22]. Specifically, for each entity h, we denote $\mathcal{N}_h = \{(h,r,t)|(h,r,t) \in \mathcal{G}_{kv}\}$ as the set of triplets, where h and t are the head and tail nodes, respectively. Then, we can perform a one-order relation graph propagation on one-hop structure to aggregate the semantic information of all tail entities,

$$\mathbf{e}_{\mathcal{N}_h} = \sum\nolimits_{(h,r,t)\in\mathcal{N}_h} \zeta(h,r,t) \cdot \mathbf{e}_t, \tag{2}$$

where $\mathbf{e}_t \in \mathbb{R}^d$ is the initial embedding of the tail node, where d indicates the dimension of embeddings, and $\zeta(h,r,t)$ is an attentive weight that reveals how much information will be propagated from t to h through relation r, which is implemented by a relation attention mechanism,

$$\begin{aligned} \zeta(h,r,t) &= \frac{exp(s(h,r,t))}{\sum_{(h,r',t')\in\mathcal{N}_h} exp(s(h,r',t'))}, \\ s(h,r,t) &= (\mathbf{W}_r\mathbf{e}_t)^{\mathsf{T}} tanh(\mathbf{W}_r\mathbf{e}_h + \mathbf{e}_r), \end{aligned} \tag{3}$$

where $s(h,r,t)$ denotes the attentive score, which is calculated based on the distance between $\mathbf{e}_h$ and $\mathbf{e}_t$ on the relation r, i.e., giving higher attention score and propagating more information for closer entities, $\mathbf{W}_r \in \mathbb{R}^{d\times d}$ is a trainable weight matrix. The final attention score $\zeta(h,r,t)$ is capable of suggesting which neighbors should be given more attention to capture semantic information by considering the relation r.

The one-order entity representation $\mathbf{e}_h$ can be obtained by aggregating the entity representation $\mathbf{e}_h$ and its relation representations $\mathbf{e}_{\mathcal{N}_h}$, i.e., $\mathbf{e}_h^{(1)} = f(\mathbf{e}_h, \mathbf{e}_{\mathcal{N}_h})$. Here we use GCN aggregator [8] as function f:

$$f(\mathbf{e}_h, \mathbf{e}_{\mathcal{N}_h}) = \sigma(\mathbf{W}_1(\mathbf{e}_h \oplus \mathbf{e}_{\mathcal{N}_h})), \tag{4}$$

where $\oplus$ is the element-wise addition to achieve self-loop operation, $\mathbf{W}_1 \in \mathbb{R}^{d\times d}$ is the trainable weight matrix to distill useful information for propagation, and σ ia a nonlinear transformation set as *LeakyReLU*. To stack more layers to gather the high-order semantic information, we can recursively formulate the representation of the entity h. Formally, in the l-th layer, the high-order propagation is calculated as,

$$\mathbf{e}_h^{(l)} = f(\mathbf{e}_h^{(l-1)}, \mathbf{e}_{\mathcal{N}_h}^{(l-1)}), \tag{5}$$

where the semantic neighbor representations $\mathbf{e}_{\mathcal{N}_h}^{(l-1)}$ in the $(l-1)$-th layer is defined as,

$$\mathbf{e}_{\mathcal{N}_h}^{(l-1)} = \sum\nolimits_{(h,r,t)\in\mathcal{N}_h} \zeta(h,r,t)\mathbf{e}_t^{(l-1)}. \tag{6}$$

Obviously, $\mathbf{e}_t^{(l-1)}$ is the representation of entity t obtained at the $(l-1)$-th layer, which memorizes the semantic information from its $(l-1)$-hop neighbors on the knowledge graph. Here, to avoid overfitting, we use dropout operation [18] on representations learned in semantic view, which has been shown to be effective in enhancing robustness on many neural networks works [26, 28].

In knowledge view, we can catch both low-order and high-order connectivity information by using a relation attentive way. Low-order connectivity such as $e_1 \xrightarrow{-r_1} v_1$ and $e_2 \xrightarrow{-r_2} v_1$, the information of attributes e_1 and e_2 can be aggregated into item v_1's semantics, the distance of items with same attributes will be closer. High-order connectivity like $v_1 \xrightarrow{r_1} e_1 \xrightarrow{-r_1} v_2$, the information of item v_1 can be propagated to item v_2 and explicitly encoded into v_2's embedding, thus the distance of the two items will be closer. Noteworthy, we extract the ground-truth relationship among items from knowledge graph to obtain the item representations with rich semantic information, which avoids the redundant mining of strong sequential correlations between items.

4.2 Session-View Representation Learning

The contextual relationship of items within a session is very significant for predicting the next item [10, 31]. Based on the directed session-view graph structure, the main task of session-view representation learning is to exploit the sequence relationship between items within a session to model item representations.

Due to the different transition relationships between items, the neighbors of an item may have different importance to this item in a session. Therefore, we utilize graph attention network to obtain the output features $\mathbf{e}_{v_i^s}$ for each item v_i^s by computing a linear weighted combination of the neighbors' features,

$$\mathbf{e}_{v_i^s}^s = \sum\nolimits_{(v_i^s, a_{r_{ij}}, v_j^s) \in \mathcal{N}_{v_i^s}} \xi(v_i^s, a_{r_{ij}}, v_j^s) \cdot \mathbf{e}_{v_j^s}, \tag{7}$$

where v_j^s is a neighbor of item v_i^s through edge relation $a_{r_{ij}}$, $\mathcal{N}_{v_i^s}$ denotes the one-order neighbors of item v_i in session-view graph, and $\xi(v_i^s, a_{r_{ij}}, v_j^s)$ is a attention score function to reduce the impact of noisy neighbors. Following [30], $\xi(v_i^s, a_{r_{ij}}, v_j^s)$ can be computed as follows,

$$\xi(v_i^s, a_{r_{ij}}, v_j^s) = \frac{exp(\mathbf{a}_{r_{ij}}^T \sigma(\mathbf{e}_{v_i^s} \otimes \mathbf{e}_{v_j^s}))}{\sum_{(v_i^s, a_{r_{ij}}, v_k^s) \in \mathcal{N}_{v_i^s}} exp(\mathbf{a}_{r_{ij}}^T \sigma(\mathbf{e}_{v_i^s} \otimes \mathbf{e}_{v_k^s}))}, \tag{8}$$

where $\otimes$ is element-wise product, σ is *LeakyReLU* activate function. $\mathbf{a}_{r_{ij}}$ denotes a relation vector to filter important features and r_{ij} indicates the relation type. Note we denote four relation vectors based on different relations in the session-view graph.

Considering that the initial embeddings of each item are semantically impoverished, we take the item representations learned from the knowledge-view graph as additional input for session-view representation learning. Then, we can obtain the feature $\mathbf{e}_{v_i^s}^{sv}$ of item v_i^s by incorporating the different output of session-view representation learning, i.e.,

$$\mathbf{e}_{v_i^s}^{sv} = \mathbf{e}_{v_i^s}^{si} \oplus \mathbf{e}_{v_i^s}^{sk}, \tag{9}$$

where $\mathbf{e}_{v_i^s}^{si}$ and $\mathbf{e}_{v_i^s}^{ss}$ are outputs of session-view representation learning with initial embedding and knowledge embedding as inputs, respectively. Notice $\mathbf{e}_{v_i^s}^{si}$ is the item representation of mining sequence information, while $\mathbf{e}_{v_i^s}^{sk}$ implies the transition of semantic information in the sequence.

For the next-click prediction, the contribution of items within a session usually is different. Typically, the items clicked later in the session are more representative of the current interest of the user, therefore, we add reversed position information to the embedding of each item [30]. Specifically, we define the position information as a learnable position embedding matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_l]$, where $\mathbf{p}_i \in \mathbb{R}^d$ is a position vector at position i. By integrating the reversed position embedding $\mathbf{p}_{l-i+1}$ into item embeddings $\mathbf{e}_{v_i^s}^{sv}$, we can obtain the position-aware item representation $\mathbf{h}_{v_i^s}$ as,

$$\mathbf{h}_{v_i^s} = tanh(\mathbf{W}_2[\mathbf{e}_{v_i^s}^{sv}; \mathbf{p}_{l-i+1}] + \mathbf{b}_1), \tag{10}$$

where $[;]$ is a concatenation operation, $\mathbf{W}_2 \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_1 \in \mathbb{R}^d$ are learnable weight matrix and bias. Obviously, the item representation $\mathbf{h}_{v_i^s}$ is an agglomeration of semantic information, sequence information, and position information.

4.3 Pairwise-View Representation Learning

As mentioned earlier, items within a session usually have feature correlations, and mining such feature-level commonality is beneficial for fine-grained session representation. Such pairwise relational information is sequence-independent. A straightforward way is to obtain the feature commonality of items by calculating the cross features $\mathbf{z}_{ij}$ for every item-item pair on the pairwise-view graph,

$$\mathbf{z}_{ij} = \mathbf{e}_{v_i^s} \otimes \mathbf{e}_{v_j^s}, \tag{11}$$

where $\mathbf{e}_{v_i^s}$ is the initial embedding of item v_i^s, $\otimes$ is the element-wise multiplication to highlight feature dimensions with similar values between two item embeddings. Then, the session representation $\mathbf{s}_{pair}$ can be calculated by performing pairwise item aggregation,

$$\mathbf{s}_{pair} = LeakyReLU(\frac{1}{l} \sum_{i=1}^{l} \sum_{j=i+1}^{l} \mathbf{z}_{ij}), \tag{12}$$

where *LeakyReLU* function is used to further filter the irrelevant common features.

However, this computational process is time-consuming with time complexity $O(l^2)$. We further simplify the computation as,

$$\begin{aligned}
\mathbf{s}_{pair} &= LeakyReLU\left(\frac{1}{l}\sum_{i=1}^{l}\sum_{j=i+1}^{l} \mathbf{e}_{v_i}^s \otimes \mathbf{e}_{v_j}^s\right) \\
&= LeakyReLU\left(\frac{1}{2l}\left(\sum_{i=1}^{l}\sum_{j=1}^{l} \mathbf{e}_{v_i}^s \otimes \mathbf{e}_{v_j}^s - \sum_{i=1}^{l} \mathbf{e}_{v_i}^s \otimes \mathbf{e}_{v_i}^s\right)\right) \\
&= LeakyReLU\left(\frac{1}{2l}\left(\sum_{i=1}^{l} \mathbf{e}_{v_i}^s \sum_{j=1}^{l} \mathbf{e}_{v_j}^s - \sum_{i=1}^{l}\left(\mathbf{e}_{v_i}^s\right)^2\right)\right) \\
&= LeakyReLU\left(\frac{1}{2l}\left(\left(\sum_{i=1}^{l} \mathbf{e}_{v_i}^s\right)^2 - \sum_{i=1}^{l}\left(\mathbf{e}_{v_i}^s\right)^2\right)\right),
\end{aligned} \tag{13}$$

where, the former term represents the square of the sum of item features within the session, and the latter term represents the sum

Table 1: Statistics of three public datasets.

Datasets		ASoftware	Yelp	Cosmetics
Sessions	#Clicks	29,455	253,975	1,219,906
	#Train	9,838	75,904	162,030
	#Test	3,055	9,995	47,951
	#Items	21,664	27,097	41,374
	Avg. length	2.30	2.67	5.34
KG	#Entities	58,367	29,082	42,101
	#Relations	4	17	2
	#Triplets	214,681	180,263	65,038

of the square of item features. Thus, the pairwise item aggregation can thus be computed in linear complexity $O(l)$. Such a graph aggregation can highlight the feature commonality of item-item pairs and improve the quality of session representation in a fine-grained pattern.

4.4 Ensemble and Prediction

We first fuse item representations from the session-view graph by a soft-attention mechanism,

$$\mathbf{s}_{sess} = \sum\nolimits_{i=1}^{l} \beta_i \cdot \mathbf{e}^{sv}_{v_i^s}, \tag{14}$$

where β_i is a attention weight associated with position-aware embeddings $\mathbf{h}_{v_i^s}$, which can be calculated as,

$$\beta_i = \mathbf{q}^T \sigma(\mathbf{W}_3 \mathbf{h}_{v_i^s} + \mathbf{W}_4 \bar{\mathbf{s}}^{sv} + \mathbf{b}_2), \tag{15}$$

where $\mathbf{W}_3$, $\mathbf{W}_4 \in \mathbb{R}^{d\times d}$ and $\mathbf{q}$, $\mathbf{b}_2 \in \mathbb{R}^d$ are separately trainable parameters, $\bar{\mathbf{s}}^{sv}$ indicates the average of item representations of the current session, i.e., $\bar{\mathbf{s}}^{sv} = \frac{1}{l}\sum_{i=1}^{l} \mathbf{e}^{sv}_{v_i^s}$. Then, we integrate two embeddings $\mathbf{s}_{sess}$ and $\mathbf{s}_{pair}$ to get final session representation $\mathbf{s}_{final}$,

$$\mathbf{s}_{final} = \mathbf{s}_{sess} \oplus \mathbf{s}_{pair}. \tag{16}$$

Obviously, the representation $\mathbf{s}_{final}$ is constructed by exploiting and fusing multiple relationships between items, including rich semantic information, contextual transition information, and pairwise commonality information. Finally, we compute the predicted probability $\hat{y}_i$ for each candidate item $v_i \in \mathcal{V}$ by dot product operation,

$$\hat{y}_i = \frac{exp(\mathbf{s}_{final}{}^\top \mathbf{e}_{v_i})}{\sum_{v_j \in \mathcal{V}} exp(\mathbf{s}_{final}{}^\top \mathbf{e}_{v_j})}, \tag{17}$$

where $\mathbf{e}_{v_i}$ indicates the trained initial embedding of item v_i. The loss function is defined as the cross-entropy of the prediction $\hat{y}_i$ and the ground-truth y_i,

$$\mathcal{L} = -\sum\nolimits_{i=1}^{n} y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i), \tag{18}$$

where $y_i = 1$ denotes item v_i appears in next item prediction, $y_i = 0$ otherwise.

5 EXPERIMENT

5.1 Experiment Setting

5.1.1 ***Dataset***. To evaluate the effectiveness of our proposed model, we conduct the experiments on three real-world datasets: Amazon software (**ASoftware** for short), **Yelp**, and **Cosmetics**, which are publicly accessible and vary in terms of domain, size and sparsity. Besides, these datasets have fundamental metadata or some properties to construct knowledge graphs.

ASoftware[1]: Amazon-review is a widely used benchmark for product recommendation [12]. We choose the subsets, Amazon software, to evaluate our proposed model.

Yelp[2]: This dataset is adopted from the 2018 edition of the Yelp challenge. Here we view the local businesses like restaurants and bars as items. Here we choose the first 1000000 interactions as the dataset for time performance consideration.

Cosmetics[3]: This dataset is a Kaggle competition dataset, which records user behaviors in a medium cosmetics online store. We use one month (October 2019) records and only retain the interactions with the type 'purchase' in our work.

5.1.2 ***Session Setup***. Cosmetics is a benchmark dataset, in which the session has been segmented. For other two datasets, considering the real-world scenarios of the datasets (i.e., the amount of software or bars that people consume in a short period of time is small), we divide each 30-day interaction into a session for ASoftware and Yelp. Afterwards, following previous works [1, 10, 31, 36], we filter out all sessions of length 1 and items appearing less than 5 times in all datasets. Furthermore, we generate sequences and corresponding labels by splitting the input sequence [20, 31]. For example, for an input session $\mathcal{S} = \{v_1^s, v_2^s, ..., v_l^s\}$, we generate a series of sequences and labels $(\{v_1^s\}, v_2^s)$, $(\{v_1^s, v_2^s\}, v_3^s)$,...,$(\{v_1^s, v_2^s, ..., v_{l-1}^s\}, v_l^s)$, where $\{v_1^s, v_2^s, ..., v_{l-1}^s\}$ is the sequence and v_l^s indicates the next-clicked item, i.e., label of the sequence.

5.1.3 ***Knowledge Graph Constructing***. We utilize metadata of ASoftware to construct its knowledge graph. For Yelp and Cosmetics, we construct their knowledge graph by extracting the properties in origin datasets. To be specific, there are 4 relations (e.g., category, brand) in knowledge graph of ASoftware, 2 relations (e.g., category and brand) in Cosmetics and 17 relations (e.g.,stars, location) in Yelp. The details of datasets are summarized in Table 1.

5.2 Baselines

We compare the proposed KMVG with the following three groups of baselines:

RNN-based SBR methods:

- **GRU4Rec** [5] is a classical RNN-based method to model the sequential pattern of items in a session by using a Gated Recurrent Unit [2].
- **NARM** [10] employs RNN with attention mechanism to capture user's preferences.

GNN-based SBR methods:

- **SR-GNN** [31] employs gated graph neural networks to capture the contextual transition between items within a session.
- **GC-SAN** [33] applies a self-attention layer after the GNN module to integrate the contextual information as session representations.
- **NISER+** [4] utilizes dropout and L_2 norm to alleviate overfitting and long-tail effect.

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] https://www.yelp.com/dataset/
[3] https://www.kaggle.com/mkechinov/ecommerce-events-history-in-cosmetics-shop

Table 2: Performance comparison(#%) on three datasets. Best performance is in boldface. Gain is obtained between BiDVAE and the best result (underscore) in baselines. $*$ indicates that the improvement is significant with $p < 0.05$.

Dataset	Metrics	GRU4Rec	NARM	SR-GNN	GC-SAN	NISER+	LESSR	MSGIFSR	CSRM	GCE-GNN	SR-GNN+KG	KM-GNN	%Gain
ASoftware	HR@20	31.07	25.34	25.26	17.45	19.07	17.47	28.03	29.60	32.37	26.57	**39.32***	21.47%
	NDCG@20	17.82	17.27	15.71	9.87	12.83	12.09	18.48	19.10	18.05	17.23	**20.14***	5.45%
	HR@10	27.23	21.45	21.56	15.25	16.59	15.94	23.46	24.68	27.02	22.34	**32.85***	21.58%
	NDCG@10	17.40	16.28	14.77	9.32	12.20	11.53	16.15	17.57	16.71	16.15	**18.57***	5.69%
Yelp	HR@20	21.62	25.59	27.77	24.56	25.86	28.66	32.34	25.22	28.69	28.04	**33.59***	3.86%
	NDCG@20	8.37	12.99	13.16	11.30	13.16	13.63	15.77	12.32	14.12	13.64	**16.89***	7.10%
	HR@10	15.84	18.81	19.61	16.64	19.08	20.00	22.90	15.89	20.58	20.23	**24.71***	7.90%
	NDCG@10	7.74	11.28	11.11	9.31	11.45	11.48	13.39	9.15	12.00	11.69	**14.68***	9.63%
Cosmetics	HR@20	39.25	43.26	44.83	44.08	45.72	45.60	46.53	45.88	46.26	44.96	**47.01***	1.03%
	NDCG@20	17.94	25.05	26.23	25.26	26.06	26.56	26.54	26.21	26.81	26.31	**27.24***	1.60%
	HR@10	32.23	35.94	37.26	36.46	37.85	37.94	38.79	34.88	38.69	37.50	**39.14***	0.90%
	NDCG@10	17.24	23.19	24.35	23.33	24.07	24.62	24.58	22.49	24.89	24.42	**25.25***	1.45%

- **LESSR** introduces shortcut graph attention and edge-order preserving aggregation layers to tackle the information loss and long-range dependency problems.
- **MSGIFSR** [3] adopts a heterogeneous session graph to extract user's multi-granularity intent for enhancing the recommendation performance.

Cross-Session SBR methods:

- **CSRM** [14] utilizes the memory networks to incorporate the latest m sessions for a better predicting of the intent of current session.
- **GCE-GNN** [30] treats items as the minimum granularity and exploits the item-transitions from session graph and global graph to model the session representations.
- **SR-GNN+KG** is an improved version by adding knowledge graph embedding on SR-GNN in the same way as KMVG.

5.3 Parameter Setting

For each dataset, we randomly select 80% of sessions to constitute the training set, and treat the remaining as the test set. Moreover, we randomly select 10% of sessions in the training set as validation set to tune hyper-parameters. For a fair comparison, we employ grid search to find the best hyper-parameters of all models based on the validation set. All parameters are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. The mini-batch Adam optimizer is exerted to optimize these parameters, where the initial learning rate is set to 0.001 and will decay by 0.1 after every 3 epochs. Moreover, the embedding dimension, the batch size and the $L2$ penalty is set to 50, 100 and 10^{-5} respectively. Specially, for GNN-based models, the number of layers is searched in 1, 2, 3, 4, and the dropout ratio is searched in 0.1, 0.2, ..., 0.9. To evaluate all models, we adopt two widely used ranking based metrics: HR@N (Hit Rate) and NDCG@N (Normalized Discounted Cumulative Gain) with the truncated list [10, 20].

5.4 Performance Comparison

Overall performance comparison of all methods are represented in Table 2. From the results, we have the follow key observations:

- *GNN-based methods generally outperform RNN-based methods on long sequential datasets.* The main reason is that RNN cannot accurately capture long-distance dependencies, while GNN excel at mining the contextual features between items within a session for next item prediction.
- *CSRM and GCE-GNN, both cross-session methods, achieve better performance than normal GNN-based methods.* For instance, GCE-GNN achieves 14.89%, 7.29% and 2.21% improvement over SR-GNN on the three datasets in term of NDCG@20, respectively. The results indicates the effectiveness of cooperating item transitions from other sessions. Besides, CSRM outperforms GCE-GNN on NDCG@20 in ASoftware, possibly because the long-term memory reused by memory networks may be beneficial on sparse ASoftware.
- *Methods of introducing knowledge graph outperforms its counterparts.* In particular, SR-GNN+KG achieve better performance than SR-GNN over all datasets and metrics, which proves the effectiveness of knowledge graph in introducing real external semantic information to enhance item representations.
- *KMVG consistently outperforms all baselines on all three datasets.* Specifically, KMVG outperforms the strongest baselines *w.r.t* HR@20 by 21.47%, 3.86%, and 1.03% on ASoftware, Yelp and Cosmetics, respectively. The results demonstrate that mining and exploiting multiple dependencies among items can improve the ability of conversational representation to facilitate recommendation. Furthermore, KMVG performs better in sparser sessions, probably because additional supplementary information (i.e., semantic information and commonality among items) becomes particularly important in the absence of sufficient transition information. For Cosmetics, less ground-truth relations between items in the knowledge graph also negatively impact further performance improvements.

5.5 Ablation Studies

We further conduct ablation experiments to investigate the effect of different components in KMVG. Table 3 shows the results in term of HR@20 and NDCG@20 on three variants: **w/o KV** is a variant of KMVG by removing semantic-view representation learning

Table 3: Impact of different components in KMVG.

Dataset	ASoftware		Yelp		Cosmetics	
Metrics	HR	NDCG	HR	NDCG	HR	NDCG
w/o KV	32.82	16.36	33.05	16.73	46.90	26.93
w/o SV	36.43	20.08	26.92	12.70	34.89	17.26
w/o PV	38.77	19.68	33.18	16.78	46.91	27.21
KMVG	**39.32**	**20.14**	**33.59**	**16.89**	**47.01**	**27.24**

and only reserve the session-view and pairwise-view. We abandon session-view in KMVG to construct **w/o SV**, which only utilize the additional information from knowledge-view and pairwise-view. Similarly, the variant **w/o PV** remove the pairwise-view representation learning and only retain knowledge-view and session-view in KMVG. From the results, we can observe the following:

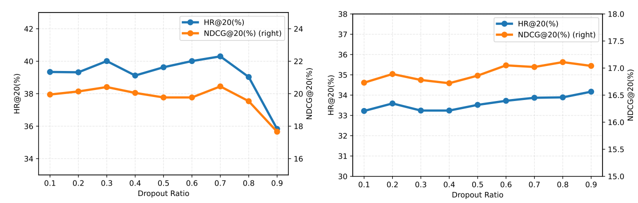
- The performance will decrease if any of the views is removed, which suggest the effectiveness of modeling session representations from multiple views and the ability of KMVG to leverage them effectively.
- NDCG@20 drops by an average of 30.37% on the three datasets after removing session-view and the performances of **w/o SV** on Yelp and Cosmetics are the worst compared to the other two variants, showing the indispensability of contextual transition in session-view graph.
- Compared with the session view, the knowledge view (**KV**) has a greater impact on ASoftware. In particular, the HR@20 on ASoftware decreased by 19.8%. The main reason is that the sparsity of ASoftware (i.e., the shortest average length in per session) is very tricky to mine accurate contextual sequence information from the session view, knowledge graphs can provide additional richer semantic information to alleviate such data sparsity. Besides, the triplet sparsity in the knowledge graph also influence the performance of model, the triples of the knowledge graph of ASoftware are the most dense in three datasets.
- Although pairwise view (**PV**) has the least impact, the performance also decreases when pairwise view is removed, indicating that mining the commonality of items in the same session is also beneficial for further enhancing session-based recommendation.

5.6 Depth Analysis of KMVG

5.6.1 ***Impact of number of layers.*** We investigate the influence of the number of layers L in knowledge view by searching L in the range of 1,2,3,4. Specifically, $L = 1$ means items only aggregate its attributes entities semantic knowledge, while $L = 2$ implies that semantic information between items with the same attribute can be transferred to each other. Table 4 reports the results of different number of layer on ASoftware and Yelp. From the reports, we can observe that $L = 2$ works best on Yelp on both metrics, while KMVG achieve best performance with $L = 1$ in term of NDCG@20 on ASoftware. The main reason may be that the knowledge-view graph of ASoftware has richer attribution entity information than Yelp, thereby aggregating insufficient attribute information is not enough for Yelp. The results demonstrate that the semantic information translation among item is beneficial for improving item representation on sparse data, while aggregation of multi-hop neighbor knowledge may be noisy and negatively affect performance.

Table 4: Impact of number of layers in knowledge-view graph.

Datasets	ASoftware		Yelp	
Measures	HR@20	NDCG@20	HR@20	NDCG@20
Layer-1	37.98	**20.23**	32.61	16.44
Layer-2	39.32	20.14	**33.59**	**16.89**
Layer-3	**39.68**	19.96	33.17	16.77
Layer-4	38.77	19.47	33.58	16.82



(a) ASoftware (b) Yelp

Figure 3: Impact of dropout ratio.

5.6.2 ***Impact of dropout ratio.*** We employ dropout regularization techniques in knowledge-view representation learning to improve the robustness of our model in training process. We explore the effect of different dropout ratio in training process by searching the dropout ratio in range of $0.1, 0.2, \ldots, 0.9$. The results on Cosmetic show similar trends with Yelp. Due to space concerns, we only present the results on ASoftware and Yelp, as shown in Figure 3. From the results, we can observe that the performances show a tortuous rise (until it reaches the peak at dropout = 0.7) and then begin to fall on ASoftware, while the performances on Yelp2018 present a tortuous upward as a whole. The reason is that there is noise in the knowledge graph. The larger the dropout ratio, the more robust the model, however too large dropout ratio will reduce the utilization of semantic relations in the knowledge graph, resulting in performance degradation, especially for ASoftware dataset with less relations in knowledge graph.

6 CONCLUSION

In this paper, we proposed a novel graph-based SBR model KMVG to mine global and local item-item relations to enhance session representations from three views, i.e., *knowledge view*, *session view* and *pairwise view*, in which tailor-made graph networks are adopted to mine item and session representations from the three views. Specifically, we adopt a knowledge graph attention network to capture the semantic information on the knowledge-view graph. For session view, we take initial embedding and knowledge embedding of items as inputs to extract complex contextual transitions of items by applying a graph attention network. Besides, we develop a pairwise item aggregator to extract commonalities between any two items of the same session as sequence-independent local item-item relations from the pairwise-view graph. Finally, the session representations learned from multiple views are fused to perform next item prediction. Comprehensive experiments on three public datasets demonstrate the superiority of KMVG over state-of-the-art methods. Further results validate the effectiveness of extracting global and local item-item relationship from our proposed three views. In future work, we will explore knowledge graph distillation to reduce the impact of unrelated entity noise, and other item-item relationships with realistic semantics are also worth questing.

REFERENCES

[1] 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.

[2] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[3] 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.

[4] Priyanka Gupta, Diksha Garg, Pankaj Malhotra, Lovekesh Vig, and Gautam M Shroff. 2019. NISER: normalized item and session representations with graph neural networks. *arXiv preprint arXiv:1909.04276* (2019).

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

[6] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.

[7] Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*. 306–310.

[8] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[9] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[10] 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.

[11] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

[12] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of EMNLP-IJCNLP*. 188–197.

[13] 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.

[14] 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.

[15] 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.

[16] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. 2001. Item-Based Collaborative Filtering Recommendation Algorithms. In *10th International World Wide Web Conference*.

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

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[19] Jianing Sun, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, Xiuqiang He, Chen Ma, and Mark Coates. 2020. Neighbor interaction aware graph convolution networks for recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1289–1298.

[20] 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.

[21] Trinh Xuan Tuan and Tu Minh Phuong. 2017. 3D convolutional networks for session-based recommendation with content features. In *Proceedings of the eleventh ACM conference on recommender systems*. 138–146.

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

[23] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 968–977.

[24] 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.

[25] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

[26] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[27] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*. 878–887.

[28] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The world wide web conference*. 2022–2032.

[29] Ze Wang, Guangyan Lin, Huobin Tan, Qinghong Chen, and Xiyang Liu. 2020. CKAN: collaborative knowledge-aware attentive network for recommender systems. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 219–228.

[30] 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.

[31] 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.

[32] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui. 2021. Self-Supervised Graph Co-Training for Session-based Recommendation. In *Conference on Information and Knowledge Management*.

[33] 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.

[34] Rui Ye, Qing Zhang, and Hengliang Luo. 2020. Cross-Session Aware Temporal Convolutional Network for Session-based Recommendation. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 220–226.

[35] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 353–362.

[36] Xiaokun Zhang, Bo Xu, Liang Yang, Chenliang Li, Fenglong Ma, Haifeng Liu, and Hongfei Lin. 2022. Price DOES Matter! Modeling Price and Interest Preferences in Session-based Recommendation. *arXiv preprint arXiv:2205.04181* (2022).