Causality and Correlation Graph Modeling for Effective and Explainable Session-based Recommendation

Cong Geng
RIIS & SIME
Shanghai University of Finance and Economics
Shanghai, China
gcong.leslie@gmail.com

Huizi Wu
RIIS & SIME
Shanghai University of Finance and Economics
Shanghai, China
wuhuizisufe@gmail.com

Hui Fang
RIIS & SIME
Shanghai University of Finance and Economics
Shanghai, China
fang.hui@mail.shufe.edu.cn

Abstract

Session-based recommendation which has been witnessed a booming interest recently, focuses on predicting a user's next interested item(s) based on an anonymous session. Most existing studies adopt complex deep learning techniques (e.g., graph neural networks) for effective session-based recommendation. However, they merely address *co-occurrence* between items, but fail to well distinguish *causality* and *correlation* relationship. Considering the varied interpretations and characteristics of causality and correlation relationship between items, in this study, we propose a novel method denoted as CGSR by jointly modeling causality and correlation relationship between items. In particular, we construct cause, effect and correlation graphs from sessions by simultaneously considering the false causality problem. We further design a graph neural network-based method for session-based recommendation. Extensive experiments on three datasets show that our model outperforms other state-of-the-art methods in terms of recommendation accuracy. Moreover, we further propose an explainable framework on CGSR, and demonstrate the explainability of our model via case studies on Amazon dataset.

***K*eywords** session-based recommendation, graph neural network, product relationship

1 Introduction

Session-based recommendation (SR) has attracted wide attention in recent years [1]. In contrast to traditional recommendation modeling users' static preferences, it processes time-aware user-item interactions to capture the dynamic preferences. Its task is to recommend next item a user will probably like given an anonymous session. Quite a series of models have been proposed to improve the performance of SR, ranging from the early Markov Chain-based ones [2] to the recent deep learning-based ones, including recurrent neural network (RNN)-based [3, 4], attention mechanism-based [5, 6] and graph neural network (GNN)-based methods [7, 8].

Although some algorithms, especially the GNN-based ones, have obtained encouraging improvements as reported, they still suffer from the following limitations: (1) RNN-based and attention-based methods focus on the dependency relationship of items within a session, but fail to easily capture item transitions across sessions; (2) GNN-based methods alleviate the aforementioned issue by constructing session graph across sessions, but they mainly model *correlation* relationship (namely co-occurrence) between items. Thus, similar to RNN-based and attention-based methods, they neglect to well distinguish directed *causality* relationship from undirected correlation relationship between items.

It is well known that *correlation* does not imply *causality* in recommendation. *Correlation* means a kind of more general, undirected relationship, i.e., two items are purchased or consumed together. In contrast, *causality* refers to a type of more directed relationship, and relates to both cause and effect where in recommendation the cause item is partly responsible for the effect item, meanwhile the effect is partly dependent on the cause (cause→effect) [9, 10]. For example, we can easily observe this kind of directed cause→effect relationship between items in real-world applications. Figure 1 illustrates three examples mined from Amazon dataset[1] [11]. As can be viewed, the number of cases that firstly buy griddlers (or water bottles/GPS navigators) and then griddler waffle plates (or capCAP/garmin portable friction mount) is much higher than that of firstly buying griddler waffle plates (or capCAP/garmin portable friction mount) followed by griddlers (or water bottles/GPS navigators). The specific statistics are 108 vs 0, 31 vs 2, and 99 vs 5, respectively.

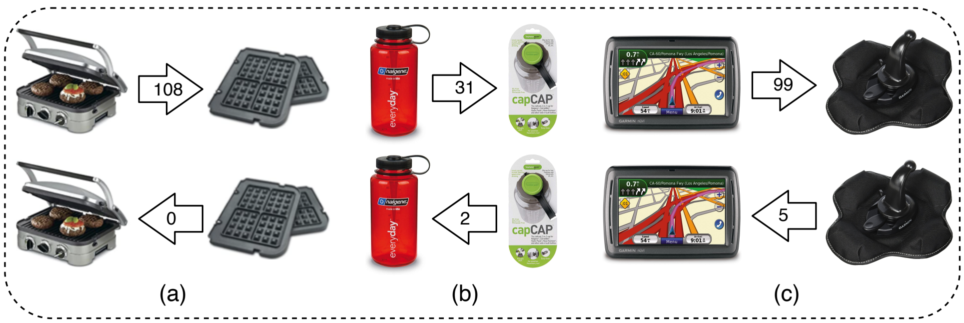


Figure 1: (a) griddlers and griddler waffle plates; (b) water bottles and capCAP; (c) GPS navigators and garmin portable friction mount.

Besides, we conduct a more comprehensive analysis to examine how to distinguish the causality and correlation between any two items on a typical session-aware dataset[2], i.e., Diginetica. In Figure 2, $p(a|b)$ ($a \neq b$) means, in a session, the probability of item a that is interacted given the previously interacted item b. It is particularly calculated as $p(a|b) = \frac{\#(b\rightarrow a)}{\#b\rightarrow *}$, where $\#(b \rightarrow a)$ is the number of item b interacted before item a in the same session, and $\#b \rightarrow *$ is the frequency of item b occurred before all other items. $p(b|a)$ is calculated in the same way. We further rank $p(a|b)$ in ascending order, and then divide all the item pair (a, b) into ten groups ($\{1, 2, \cdots, 10\}$). Thus, each group has the same number of item pairs, and from group 1 to 10, $p(a|b)$ gets bigger. We deal with $p(b|a)$ similarly. Finally, we can place each item pair (a, b) into a grid in terms of $p(a|b)$ and $p(b|a)$ as shown in Figure 2.

For example, $33,758$ in grid $(4, 1)$ in Figure 2 refers to the number of item pairs that $p(a|b)$ and $p(b|a)$ belong to the corresponding group 4 and 1, respectively. In this case, if $|p(a|b) - p(b|a)| \geq \epsilon$ (ϵ is a non-zero value), we consider that the relationship between items a and b is asymmetrical (i.e. sort of directed), and a larger ϵ indicates a more directed relationship between items a and b. Accordingly, the relationship between item pairs fallen into the bottom right (upper left) of Figure 2 is more asymmetrical, revealing a higher possibility for being in the causal relations. Furthermore, the relationship fallen into the upper right implies much stronger undirected correlation relationship (i.e., less possible causal relations). Both of the directed and un-directed relations are quite prevalence (similar patterns can be viewed on Gowalla[3] and Amazon Home and Kitchen in Figure 3), in this case, we should carefully consider both correlation and causality between items in item modeling for recommendation.

Although some studies [12, 13, 14] consider correlation relationship implicitly, overall the directionality of cause item to effect item has been insufficiently explored [10]. In this study, we choose to define the "causality" as a kind of directional relationship, which is not equivalent to that in the traditional sense. Causality in our study refers to a direct and asymmetric relationship, which is calculated from a relatively large volume of sessions. The higher the edge weight in causality graph, the higher the probability of this directional relationship between items. In summary, our motivation is not to capture all causality relationships but to explore the relationship between items from a causality perspective to achieve more effective recommendations.

Therefore, considering the difference between the two types of item relationship, we propose a novel method called Causality and Correlation Graph Modeling for Effective and Explainable Session-based Recommendation (CGSR) by

[1] `jmcauley.ucsd.edu/data/amazon/links.html`.
[2] `competitions.codalab.org/competitions/11161#learn_the_details-data2`.
[3] `snap.stanford.edu/data/loc-Gowalla.html`.

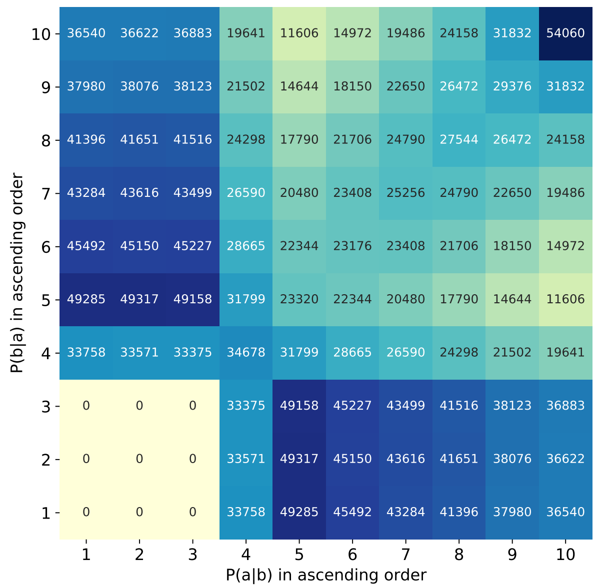
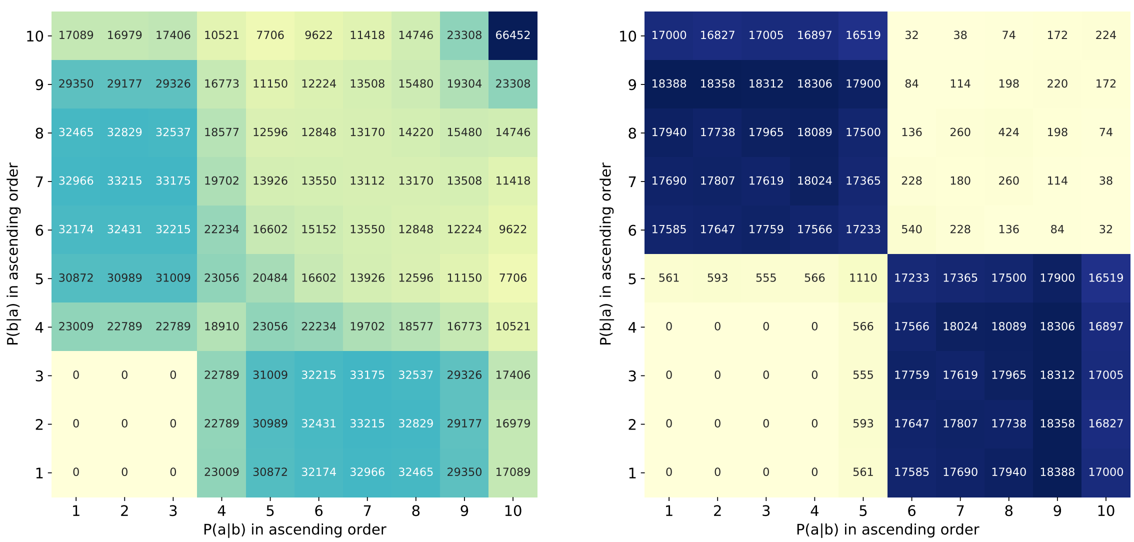


Figure 2: Causality statistics on Diginetica dataset.



(a) Gowalla

(b) Amazon

Figure 3: Causality statistics on two datasets.

particularly taking the causality and correlation relationship between items into consideration. Specifically, other than a correlation graph considering both first-order and three types of second-order relationship, we construct two graphs to capture causality relationship: a cause graph and an effect graph, which address the false causality by removing the impact of the common cause items given an item pair. Then, we design an end-to-end GNN-based model which takes the three graphs as input, outputs three kinds of item embeddings, and deploys attention mechanism to obtain corresponding session representations for final recommendation. Experimental results on three real-world datasets validate the superiority of our approach over the state-of-the-art. We further propose an explainable framework based on CGSR for SR, and conduct case studies on Amazon dataset to showcase that CGSR can also facilitate the explanation task in SR.

The main contributions of this work are four-fold:

- To the best of our knowledge, we are the first to explore causality between items for SR, and propose CGSR to enhance the recommendation and explanation tasks in SR.
- We design an effective mechanism to capture possibly directed causality relationship between items. Particularly, we construct an effect graph and a cause graph on sessions which rule out the false causality by eliminating the impact of common cause items for every item pair.
- We design a GNN-based method to combine causality and correlation graphs for effective recommendation. Regarding correlation graph, we consider both first-order and three types of second-order relationship. Exhaustive experiments on three real datasets verify the superiority of our approach over other baselines, and the validity of our designs.
- We contribute to explainable SR by figuring out an explainable framework grounded on CGSR. Case studies on Amazon dataset demonstrate its usability and feasibility.

2 Related Work

Our work is related to two primary tasks: session-based recommendation, and item relation modeling in recommendation. In the following subsections, we discuss each part to highlight our contributions over the related studies.

2.1 Session-Based Recommendation

Session-based recommendation (SR) predicts a user's next interested item(s) by deploying traditional approaches and deep learning (DL) techniques to process time-aware user-item interactions. Traditional approaches [15, 16] apply machine learning (ML) techniques to capture item embedding in the session. For example, FPMC [2] applies matrix factorization (MF) and first-order Markov chains (MC) to address the sequential relationship among items. SEQ* [17] develops a hidden Markov model for sequences preference, which considers more factors, including two types of dynamic factors and contextual factors. [18] propose a personalized Markov embedding (PME), which embeds songs and users into a Euclidean space, where the distances present the strengths of their relationships.

On the contrary, DL methods [19, 20, 21] are capable of dealing with a much longer sequence than traditional models. GRU4Rec [3] firstly applies recurrent neural network (i.e., a multi-layer gate recurrent unit) to process session data. Later, there are a lot of variants with regard to GRU4Rec. For instance, HRNNs [22] extends it to the hierarchical form which simultaneously considers both short-term and long-term preferences with two GRU constructs, i.e., the session-level GRU (GRUses) and the user-level GRU (GRUusr). Donkers et al. [23] model the temporal dynamics of consumption sequences based on the gated RNN and explicitly represent the individual user in a gated architecture. NARM [24] combines GRUs and vanilla attention mechanism to better extract main purpose from the current session, which can effectively eliminate noise from unintended behaviors. However, these methods mainly address behavior dependency in a session, but cannot directly capture item relationship across different sessions. Besides, they ignore to distinguish causality relationship from correlation relationship between items.

With the rapid development of graph neural networks (GNN) in recent years, we have witnessed its great success in many downstream tasks, e.g., node classification and recommender systems. Therefore, some studies have started to deploy GNNs for session-based recommendation, and obtained encouraging results [25, 26, 27]. For example, SR-GNN [12] firstly combines different sessions into session graphs, and then uses GNN [13] to learn representation of each item and finally obtain session representations through attention mechanism. Experimental results on Yoochoose and Diginetica verify that it can obtain better performance than both RNN-based models (e.g., GRU4Rec) and attention based models (e.g., STAMP [28]). Later, quite a few variants of SR-GNN [29, 30, 31] have been proposed. For example, Xu et al. [29] introduce a novel graph contextual self-attention model based on the graph neural network called GC-SAN, which obtains local graph structured dependencies of separated session sequences and models contextualized non-local representations. LESSR [30] further reduces information loss by proposing an edge-order preserving aggregation layer and a shortcut graph attention layer. However, as have been discussed, all GNN-based methods have not directly considered the directed causality relationship between items, which might lead to incorrect recommendations.

2.2 Relation Modeling in Recommendation

Quite a few studies target to explore various relationship between items using side information like textual and visual information [32, 33]. For example, Sceptre [34] casts the item relations identification problem as a supervised link prediction task, and predicts substitutable and complementary items by learning latent topics from textual information. Zhang et al. [33] propose a neural complementary recommender Encore which can jointly learn complementary item relationship and user preferences through Bayesian inference. However, these studies aim to design specific models for identifying the well-studied relationship (e.g., substitute and complementary) between items from side information other than interaction data. On the contrary, in our study, relying on the sequential interaction data in sessions, we try to identify the directed causality relationship between items for effective recommendation. Our causality relationship is expected to complement the identified item relationship widely discussed in the business area.

There are also some studies that have explored the causal inference for recommendation [35, 36, 37, 38]. For example, Bonner and Vasile [39] optimize the causal recommendation outcomes via user implicit feedback based on the factorizing matrices, which presents that the objective of causal recommendations is equal to factorizing a matrix of user responses. Wang et al.[40] consider that the core of recommender system is to address a causal inference questionby solving two problems: which items the users decide to interact with, and how the users rate them. Qiu et al. [41] proposes a deconfounded recommender, which utilizes Poisson factorization to infer confounders in treatment assignments. We can see that this line of research is more related to unbiased recommendation and tries to reason about personalized user preference well, which is quite different from our research scenario. In our study, we try to explore the directed causality relationship between items to facilitate session-based recommendation.

3 Graph Construction

In this section, we firstly formally define the research problem, and then present how to construct causality and correlation graphs from sessions, as well as the edge weights in great details.

3.1 Problem Statement

In session-based recommendation, let $\mathcal{V} = \{v_1, v_2, \cdots, v_N\}$ denote the set of items ($|\mathcal{V}| = N$). An anonymous session S can be represented by $S = \{v_1^S, v_2^S, \cdots, v_l^S\}$, where its length equals to l and $v_i^S \in \mathcal{V}$ means the i-th interacted item within S. There are M sessions in total. Given session S, the goal of session-based recommendation is to predict the next item (i.e., the $l+1$-th) that will be purchased. Therefore, in our CGSR, we strive to firstly construct graphs (i.e., causality and correlation graphs) from training sessions, and then learn effective representation of session S. Thirdly, we generate recommendation score $\widehat{y}_j$ for each candidate item $v_j \in \mathcal{V}$, and finally recommend top K items with the highest recommendation scores.

Next, we will elaborate causality and correlation graphs construction in detail, respectively.

3.2 Constructing Causality Graphs

While constructing causality graphs from sessions, we aim to maximally identify the truly directed causality relationship meanwhile ruling out the false ones (i.e., the noisy information). To fulfill the goal, as shown in Figure 7(a), we firstly build a *session graph* from sessions, on the basis of which, we then construct an *effect graph* and a *cause graph* by removing the impact of noisy information. Noted that, in causality graphs, we only consider first-order relationship since we strive to directly extract the most probability causal relations between items given historical data and high-order ones might involve more noise. Besides, GNN model is supposed to automatically capture high-order relations.

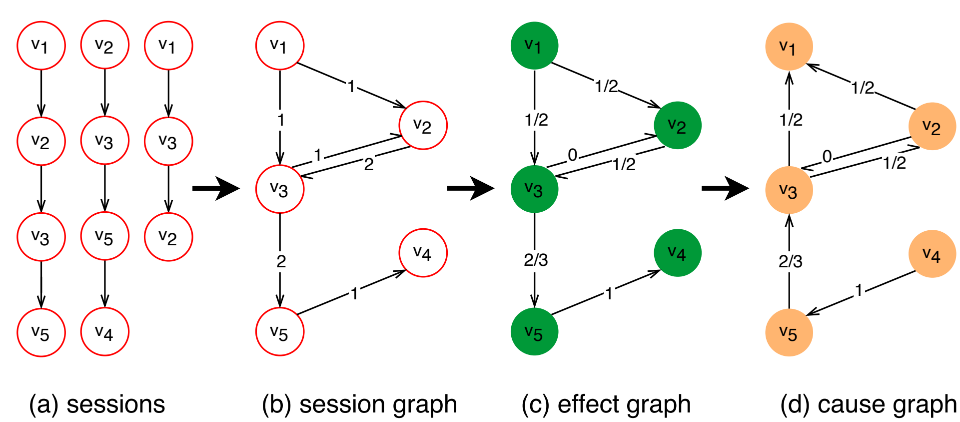


Figure 4: The process of constructing causality graphs (i.e., an effect graph and a cause graph).

Session graph construction. Without the loss of generality, let $\mathcal{G}^s = (\mathcal{V}^s, \mathcal{E}^s)$ be the correspondingly directed session graph, where $\mathcal{V}^s$ indicates the node set which is identical to the item set in all sessions, and $\mathcal{E}^s$ denotes the edge set. An edge $E^s_{i,j} \in \mathcal{E}^s$ refers to that in a session, item v_i is firstly interacted followed by item v_j, and $w^s_{i,j}$ is the corresponding occurred frequency to denote the weight of edge $E^s_{i,j}$. For example, in Figure 4, these three sessions in Figure 4 (a) can form a directed session graph in Figure 4 (b).

Causality graphs construction. To better represent the causality relationship, we construct two graphs: an effect graph ($\mathcal{G}^e$) and a cause graph ($\mathcal{G}^c$) on the basis of the session graph. Specifically, in *effect graph*, we want to learn representation of an item where the item is playing the *effect* role in the directed cause-effect relationship between two items. Consequently, each item representation is expected to integrate the information of the adjacent cause items by information propagation via GNN. In contrast, the *cause graph* is to capture the item information where the item is playing the *cause* role in the directed relationship. We clarify the following two issues: (1) the node and edge sets of the effect graph are initially the same as those of session graph, mainly because the session graph models the temporally directed relationship between items, basically indicating all the possible causality relationship from sessions. However, as $\mathcal{G}^s$ may involve false causality relationship (elaborated later), we will eliminate its impact by justifying the corresponding weight of each edge; (2) the cause graph is quite similar to the effect graph, except that the direction of each edge is opposite to that of effect graph, considering that in cause graph, we want to explore the part of item information leading to the purchase of other items.

Towards the first issue, we elaborate *how to get rid of the impact of false causality relationship*. Let us first view an example. There are three sessions: S_1 ["iPhone", "charging line", "charger", "phone shell"], S_2 ["iPhone", "charger", "charging line"], and S_3 ["iPhone", "charging line", "phone shell"]. Based on the three sessions (training data), a phone shell will be recommended given session ["charging line", "charger"], which is quite odd since few users will purchase a phone shell if having previously bought a charging line and charger. By examining the data, intuitively, we see that "owing an iPhone" is a *common cause* to also purchase a charging line, charger and phone shell. In this case, such odd recommendation is probably induced by the false causality relationship led by the common cause (i.e., "iPhone") in training data. That is, purchasing "iPhone" leads to the purchase of "charger" and "phone shell", instead of purchasing "charger" causing the purchase of "phone shell".

Therefore, to overcome this issue, we appropriately calculate the weight of each edge in $\mathcal{G}^e$ by taking the common cause of two items into consideration. Specifically, for each item pair v_i and v_j, we firstly identify every common cause $v_k \in I^s_i \cap I^s_j$ (I^s_* is the node set directed into node v_* in $\mathcal{G}^s$). For example, in Figure 4 (b), v_1 is the common cause to v_2 and v_3. Secondly, to identify the true causality strength of $v_i \to v_j$ in $\mathcal{G}^e$, we eliminate the impact of every common cause v_k to calculate the weight of $E^e_{i,i}$, $w^e_{i,j}$:

$$w^e_{i,j} = \frac{w^s_{ij} - \sum\limits_{v_k \in I^s_i \cap I^s_j} \#[v_k, v_i, v_j]}{\#[v_i, v_*]} \tag{1}$$

where $\#[v_k, v_i, v_j]$ and $\#[v_i, v_*]$ are the number of sequence $[v_k, v_i, v_j]$ and the number of sequence $[v_i, v_*]$ ($v_* \in V$) existing in all sessions (training data), respectively. For example, the weights of edges in Figure 4 (c) are computed as: $w_{1,2} = (1-0)/2$; $w_{1,3} = (1-0)/2$; $w_{2,3} = (2-1)/2$; $w_{3,2} = (1-1)/2$; $w_{3,5} = (2-0)/3$; $w_{5,4} = (1-0)/1$. After obtaining the effect graph $\mathcal{G}^e$, we reverse its directions of edges to get the cause graph $\mathcal{G}^c$.

3.3 Constructing Correlation Graph

Besides the directed causality relationship between items, we also consider the undirected correlation relationship, whose effectiveness in SR has been validated in previous GNN-based studies [12, 31]. While constructing the correlation graph, apart from the generally adopted *first-order* relationship (neighbor in sequence), we additionally consider the *second-order* relationship (neighbor of neighbor), by following the study of [31]. Noted that differing from [31], we distinguish three kinds of second-order relationship (i.e., *chain*, *fork*, and *collider*, see Figure 5) for better exploring the correlation relationship for effective recommendation.

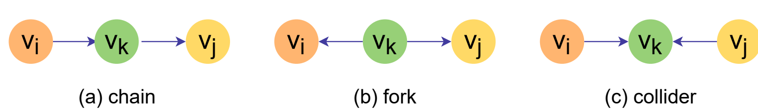
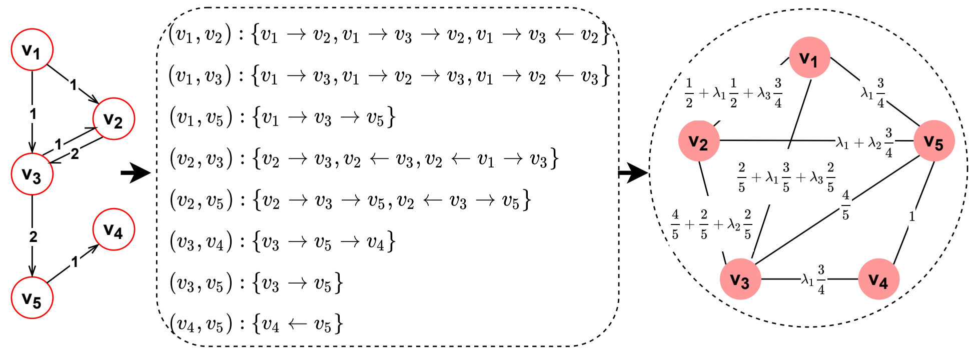


Figure 5: Three types of second-order relationship.

Particularly, as shown in Figure 6, based on the session graph $\mathcal{G}^s$ have been constructed, we calculate the weight of each possible edge by considering the first-order and second-order neighbors.



(a) session graph (b) first-order and second-order relationship (c) corrrelation graph

Figure 6: The process of constructing correlation graph.

For the first-order relationship, namely (v_i, v_j) in session graph $\mathcal{G}^s$, considering $\mathcal{G}^s$ is directed whilst correlation graph $\mathcal{G}^r$ is undirected, the first-order weight of each edge, $w_{i,j}^{r,1}$, is computed as:

$$w_{i,j}^{r,1} = \underbrace{\frac{2 * w_{i,j}^s}{\sum\limits_{v_k \in O_i^s} w_{i,k}^s + \sum\limits_{v_k \in I_j^s} w_{k,j}^s}}_{\text{impact of } v_i \to v_j} + \underbrace{\frac{2 * w_{j,i}^s}{\sum\limits_{v_k \in O_j^s} w_{j,k}^s + \sum\limits_{v_k \in I_i^s} w_{k,i}^s}}_{\text{impact of } v_j \to v_i} \tag{2}$$

where O_*^s and I_*^s refer to the node (item) set directed out and into item v_* in $\mathcal{G}^s$ respectively.

For the second-order relationship, we first extract all the possible second-order relationship for each item pair v_i and v_j based on $\mathcal{G}^s$. That is, if v_k is the neighbor of both v_i and v_j, we say that there is a second-order correlation between v_i and v_j. In particular, with the consideration on the link directions among the corresponding three items, we identify three types of second-order relationship, intuitively denoted as *chain*, *fork* and *collider* (see Figure 5). We treat the three types separately mainly because they act differently on casting correlation relationship between items [42], which are modeled as different weighting factors in our study (see λ_1, λ_2 and λ_3 in Equation 3). Thus, the weight of link (v_i, v_j) in terms of second-order correlation, $w_{i,j}^{r,2}$, is thus computed as:

$$\begin{aligned} w_{i,j}^{r,2} = & \underbrace{\lambda_1 \frac{\sum\limits_{v_k \in O_i^s \cap I_j^s} \left(w_{i,k}^s + w_{k,j}^s\right)}{\sum\limits_{v_k \in O_i^s} w_{i,k}^s + \sum\limits_{v_k \in I_j^s} w_{k,j}^s} + \lambda_1 \frac{\sum\limits_{v_k \in I_i^s \cap O_j^s} \left(w_{j,k}^s + w_{k,i}^s\right)}{\sum\limits_{v_k \in O_j^s} w_{j,k}^s + \sum\limits_{v_k \in I_i^s} w_{k,i}^s}}_{\text{chain}} \\ & \underbrace{+ \lambda_2 \frac{\sum\limits_{v_k \in I_i^s \cap I_j^s} \left(w_{k,i}^s + w_{k,j}^s\right)}{\sum\limits_{v_k \in I_i^s} w_{k,i}^s + \sum\limits_{v_k \in I_j^s} w_{k,j}^s}}_{\text{fork}} + \underbrace{\lambda_3 \frac{\sum\limits_{v_k \in O_i^s \cap O_j^s} \left(w_{i,k}^s + w_{j,k}^s\right)}{\sum\limits_{v_k \in O_i^s} w_{i,k}^s + \sum\limits_{v_k \in O_j^s} w_{j,k}^s}}_{\text{collider}} \end{aligned} \tag{3}$$

where λ_1, λ_2 and λ_3 are trainable parameters to balance the impact of the three second-order relationship types.

Finally, we get the weight of every edge $E_{i,j}^r$ in correlation graph $\mathcal{G}^r$, $w_{i,j}^r$, by adding the first-order and second-order weights, namely, $w_{i,j}^r = w_{i,j}^{r,1} + w_{i,j}^{r,2}$, as depicted in Equation 2 and 3. Noted that compared with $\mathcal{G}^s$, $\mathcal{G}^r$ can obtain new edges by the three types of second-order correlations, e.g., $E_{1,5}^r$, $E_{2,5}^r$, $E_{3,4}^r$ in Figure 6 (c).

4 The CGSR Model

In this section, we present our proposed CGSR model detailedly. Figure 7 outlines the overview of CGSR, which consists of four components: (a) *Graph Construction*, (b) *Item Representation Learner*, (c) *Session Representation Learner* and (d) *Recommendation Score Generator*. In particular, we firstly build three types of graphs (i.e., effect graph $\mathcal{G}^e$, cause graph $\mathcal{G}^c$ and correlation graph $\mathcal{G}^r$) in *Graph Construction* as introduced in Section 3. *Item representation Learner* deploys a weighted graph attention network (WGAT) on each of the three graphs to obtain a representation for each item, respectively. That is, for each item v_i, we obtain three types of representations, namely $\mathbf{x_i}^e$, $\mathbf{x_i}^c$, and $\mathbf{x_i}^r$, given $\mathcal{G}^e$, $\mathcal{G}^c$ and $\mathcal{G}^r$. *Session Representation Learner* uses an Attention Layer to aggregate each type of learned item representation in session sequence S to obtain session representation $\mathbf{S}^e$, $\mathbf{S}^c$ and $\mathbf{S}^r$, respectively. We also get session representation $\mathbf{S}^p$ by averaging the three types of session representation. *Recommendation Score Generator* strives to calculate the recommendation score, $\hat{y}_j$ of each candidate item $v_j \in \mathcal{V}$ on the basis of the learned item and session representations. Next, we elaborate the last three components of CGSR.

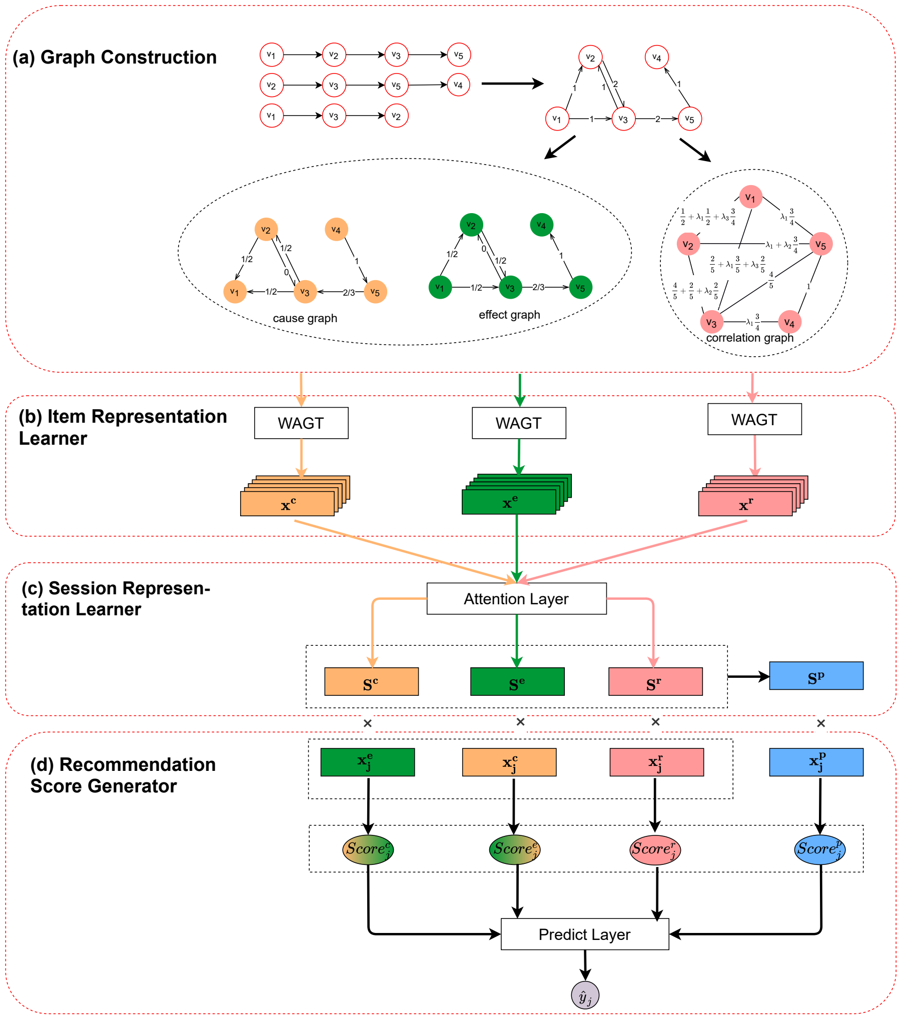


Figure 7: The overview of our proposed CGSR model.

4.1 Item Representation Learner

Here, we aim to learn item embedding on built graphs $\mathcal{G}^c$, $\mathcal{G}^e$ and $\mathcal{G}^r$. Considering that the three graphs are either weighted (correlation graph) or simultaneously directed and weighted (cause and effect graphs), we thus adopt weighted graph attention network (WGAT) [14] to obtain item representations. Specifically, we denote $\mathbf{X}^0 \in \mathbb{R}^{N \times d_0}$ as initial embedding matrix of item set (for item $v_i \in \mathcal{V}$, initial item embedding $\mathbf{x_i}^0 = \mathbf{X}^0_{i,:} \in \mathbb{R}^{d_0}$):

$$\mathbf{X}^0 = \text{nn.Embedding}(N, d_0) \tag{4}$$

Then, taking cause graph $\mathcal{G}^c$ as an example, in WGAT, self-attention mechanism is deployed to aggregate information from each node (item) v_i's directed-into neighbors. As in our scenario, a session is normally not very long, all first-order neighbors are considered. Thus, the importance between v_i and its neighbor v_j ($v_j \in I_i^c$, and I_i^c is the node set directed into v_i in $\mathcal{G}^c$) , $e^c_{i,j}$ (i.e., self-attention coefficient) is computed as:

$$e^c_{ij} = \sigma(\mathbf{W}^T_{c,2} * [\mathbf{W}_{c,1}\mathbf{x}^0_i; \mathbf{W}_{c,1}\mathbf{x}^0_j; w^c_{j,i}]) \tag{5}$$

where $\sigma(.)$ is the Leaky ReLU function, $\mathbf{W}_{c,1} \in \mathbb{R}^{d \times d_0}$ and $\mathbf{W}_{c,2} \in \mathbb{R}^{2d+1}$ are trainable parameters, and $w^c_{j,i}$ is the weight of link $v_j \to v_i$. Softmax function is further adopted to normalize the e^c_{ij}:

$$\alpha^c_{ij} = \text{softmax}(e^c_{ij}) = \frac{exp(e^c_{ij})}{\sum\limits_{v_k \in I^c_i} exp(e^c_{ik})} \tag{6}$$

Third, the information from neighbors are weighted to get item v_i's embedding, $\mathbf{x_i^c} \in \mathbb{R}^d$:

$$\mathbf{x}^c_i = \sigma(\sum_{j \in I^c_i} \alpha^c_{ij} \mathbf{W}_{c,3} \mathbf{x}^0_j) \tag{7}$$

where $\mathbf{W}_{c,3} \in \mathbb{R}^{d \times d_0}$ is a trainable parameter matrix, and $\sigma(.)$ is the Leaky ReLU function.

In multi-head attention mechanism of WGAT, we average obtained embedding from K_m heads to output final item embedding:

$$\mathbf{x_i^c} = \frac{1}{K_m} * \sum_{k=1}^{K_m} \mathbf{x^c_{i,k}} \tag{8}$$

where $\mathbf{x}^c_{i,k}$ is v_i's representation output by the k-th head using Equations 5-7.

Similarly, we get the embedding of v_i, $\mathbf{x}^e_i$ and $\mathbf{x}^r_i$, from effect graph $\mathcal{G}^e$ and correlation graph $\mathcal{G}^r$ respectively.

4.2 Session Representation Learner

In *Item Representation Leaner*, we obtain three embedding for each item v_i in terms of three graphs. Specifically, $\mathbf{x}^c_i$ considers information from v_i's effect neighbors, whilst $\mathbf{x}^e_i$ and $\mathbf{x}^r_i$ involve information from its cause and correlated neighbors, respectively.

In contrast to previous studies [31] fusing embedding from various sources to generate session representation, in CGSR, *Session Representation Learner* treats each type of item representation separately. That is, it learns a specific session representation given each type of item embedding, for the purpose of obtaining a rather pure session representation for each relationship instead of losing information by aggregating them too early.

Specifically, for instance, in terms of cause graph $\mathcal{G}^c$, to generate the representation of session S ($S = \{v^S_1, \cdots, v^S_l\}$), $\mathbf{S}^c$, using attention mechanism, we first compute the weighted factor α^c_k depicting the importance of k-th item (v^S_k) to l-th item (last item, v^S_l) in S:

$$\alpha^c_k = \mathbf{q}^T_c \sigma(\mathbf{W}_{c,4} \mathbf{x}^c_{l,S} + \mathbf{W_{c,5}} \mathbf{x}^c_{k,S} + \mathbf{b}_c) \tag{9}$$

where $\mathbf{q}_c \in \mathbb{R}^d$ denotes the weighting vector. $\mathbf{W}_{c,4}$, $\mathbf{W}_{c,5} \in \mathbb{R}^{d \times d}$ are the weighting matrices. $\mathbf{b}_c \in \mathbb{R}^d$ is the bias vector. $\sigma(.)$ is the sigmoid function. $\mathbf{x}^c_{l,S}$ and $\mathbf{x}^c_{k,S}$ denote the embedding of l-th and k-th items given cause graph respectively. We thus use weighted factors to aggregate all item information for session representation:

$$\mathbf{S}^c_g = \sum_{k=1}^{l} \alpha^c_k \mathbf{x}^c_{k,S} \tag{10}$$

Following the previous studies, we also particularly consider the information of the most recent behavior in the session (i.e., v^S_l), namely, to get the concatenation of $\mathbf{S}^c_g$ and $\mathbf{x}^c_{l,S}$. We further project the concatenation to get the final session representation via:

$$\mathbf{S}^c = \mathbf{W}_{c,6}[\mathbf{x}^c_{l,S}; \mathbf{S}^c_g] \tag{11}$$

where $\mathbf{W}_{c,6} \in \mathbb{R}^{d \times 2d}$ is the projecting matrix.

Similarly, we can learn the session representation on effect and correlation graphs, $\mathbf{S}^e$ and $\mathbf{S}^r$, respectively. We also generate a session representation ($\mathbf{S}^p$, referred as preference-related session representation) by fusing the three types of session representation using mean operator, and further project it into a new latent space:

$$\mathbf{S}^p = \mathbf{W}_7 * \text{mean}(\mathbf{S}^c, \mathbf{S}^e, \mathbf{S}^r) \tag{12}$$

where $\mathbf{W}_7 \in \mathbb{R}^{d \times d}$ is the projecting matrix. mean$(*)$ function outputs the corresponding average value.

In summary, *Session Representation Leaner* outputs four session representations of session S, i.e., $\mathbf{S}^c$, $\mathbf{S}^e$, $\mathbf{S}^r$, and $\mathbf{S}^p$.

4.3 Recommendation Score Generator

Given the learned session representations of session S, for each candidate item v_j, *Recommendation Score Generator* will output its recommendation score. Particularly, the final score is three-fold: (1) causality score; (2) correlation score; and (3) preference score.

Causality score strives to maximize item transition from cause $\rightarrow$ effect whilst simultaneously minimize that from effect $\rightarrow$ cause. Accordingly, causality score of item v_j, $Score_j^{ca}$, is calculated as:

$$Score_j^{ca} = Score_j^{c} - Score_j^{e} = (\mathbf{S}^c)^T \mathbf{x}_j^e - \gamma_1 (\mathbf{S}^e)^T \mathbf{x}_j^c \tag{13}$$

where γ_1 is a trainable parameter to balance the two components.

Correlation score and *preference score* of v_j, $Score_j^r$ and $Score_j^p$, are defined as:.

$$Score_j^r = (\mathbf{S}^r)^T \mathbf{x}_j^r; \quad Score_j^p = (\mathbf{S}^p)^T \mathbf{x}_j^p \tag{14}$$

where $\mathbf{x}_j^p = \text{mean}(\mathbf{x}_j^c, \mathbf{x}_j^e, \mathbf{x}_j^r)$.

Finally, we compute the overall score of candidate item v_j ($Score_j$):

$$Score_j = Score_j^p + \gamma_2 Score_j^{ca} + \gamma_3 Score_j^r \tag{15}$$

where γ_2 and γ_3 are trainable parameters. Softmax function is further deployed to obtain the final recommendation score $\hat{y}_j$:

$$\hat{y}_j = \text{softmax}(Score_j) = \frac{exp(Score_j)}{\sum\limits_{v_k \in \mathcal{V}} exp(Score_k)} \tag{16}$$

We adopt cross-entropy loss to train the CGSR model:

$$L = -\sum_{j=1}^{N} y_j \log(\hat{y}_j) + (1 - y_j) \log(1 - \hat{y}_j) \tag{17}$$

where y_j is the ground-truth recommendation score of item v_j.

5 Empirical Evaluations

In this section, we conduct extensive experiments on three datasets to validate the effectiveness of our proposed CGSR, with the goal of answering four specific research questions (RQs):

- **RQ1**: How does CGSR perform compared to other state-of-the-art approaches?
- **RQ2**: How do different components of CGSR (e.g., causality graphs) contribute to the recommendation performance?
- **RQ3**: How do different hyper-parameters affect the performance of CGSR?
- **RQ4**: How does CGSR facilitate the explanation task of session-based recommendation?

5.1 Experimental Setup

5.1.1 Datasets.

We choose three real-world datasets, i.e. Diginetica, Gowalla, and Amazon home and kitchen, which (especially the first two) are commonly used in session-based recommendation, to evaluate the performance of different approaches. In particular, *Diginetica* is from CIKM cup 2016 and consists of typical transactions data. Following [24, 28, 14, 12, 43, 30], we filter out items with less than 5 interactions, and sessions with length smaller than 2. Besides, we use the sessions occurred in the last week as the test set. *Gowalla* is a check-in dataset for point-of-interest (POI) recommendation. Following [30, 44, 45], we keep the top $30,000$ most popular items, and treat a user's check-ins within a day as a session. Besides, we filter out sessions with length smaller than 2 or larger than 20. We sort sessions with increasing timestamp and use the last 20% of all the sessions as the test set. *Amazon* contains product reviews and metadata from Amazon in home and kitchen category. We consider a user's interactions occurred in a day as a session, and filter out items with less than 5 interactions and sessions with length smaller than 2. Besides, we use the last 20% sessions as test data. The statistics of the three datasets are summarized in Table 1.

Table 1: Statistics of the three datasets.

Dataset	Diginetica	Gowalla	Amazon
#transactions	982,961	1,122,788	335,639
#items	43,097	29,510	38,689
#sessions	780,328	830,893	246,661
average length	5.12	3.85	3.77
#train sessions	719,470	675,561	198,861
#test sessions	60,858	155,332	47,800

5.1.2 Baseline Models.

We compare our CGSR with three traditional methods (**POP**, **ItemKNN** and **FPMC**), two RNN-based methods (**NARM** and **GRU4Rec**), and three state-of-the-art (SOTA) GNN-based methods (**SR-GNN**, **FGNN** and **LESSR**) for session-based recommendation.

- **POP** recommends the most popular items in the training set;
- **ItemKNN** [46] recommends items having the highest similarity with the last item of the session;
- **FPMC** [2] combines matrix factorization with the first-order MCs;
- **GRU4Rec** [3] stacks GRUs to process session data and tailors an ranking loss function to train the model;
- **NARM** [24] is a strong and solid RNN-based approach for SR, which utilizes vanilla attention to model the relationship of the last item with other items in a session to capture the main purpose;
- **SR-GNN** [12] uses a gated graph convolutional layer to obtain item embedding on session graph, and applies self-attention mechanism to last item embedding to obtain session representation;
- **FGNN** [14] designs a novel model to collaboratively incorporate sequence order and latent order in the session graph;
- **LESSR** [30] proposes a lossless encoding scheme and an edge-order preserving aggregation layer based on GRU, and designs a shortcut graph attention layer to effectively capture long-range dependencies among items.

5.1.3 Evaluation Metrics.

We adopt three widely used ranking-based metrics: **HR**@K, **MRR**@K and **NDCG**@K[4] to evaluate the recommendation accuracy, where a higher value indicates better performance, and K is set to 5, 10 and 20, respectively. **HR**@K (Hit Ratio) denotes the hit ratio, i.e., the coverage rate of targeted predictions; **MRR**@K (Mean Reciprocal Rank) indicates the ranking accuracy based on the ranking position of the recommended items (hits), and a larger value means the ground-truth items are ranked in the top of the ranked recommendation lists; **NDCG**@K (Normalized Discounted Cumulative Gain) also rewards each hit based on its position in the ranked recommendation list.

Table 2: Hyper-parameter setups of baselines.

Method	**Datasets**	**Hyper-parameter setups**
GRU4Rec	Diginetica, Gowalla, Amazon	GRU size=100, Batch size=32, Lr=0.2
NARM	Diginetica, Gowalla, Amazon	Embedding size=50, Batch size=512, Lr=0.001
SR-GNN	Diginetica, Gowalla	Embedding size=100, Batch size=100, Lr=0.001, L_2 penalty=1e-5
	Amazon	Embedding size=170, Batch size=100, Lr=0.001, L_2 penalty=1e-5
FGNN	Diginetica, Gowalla	Embedding size=100, Batch size=100, Lr=0.001, L_2 penalty=1e-5
	Amazon	Embedding size=150, Batch size=100, Lr=0.001, L_2 penalty=1e-5
LESSR	Diginetica	Embedding size=32, Batch size=512, Lr=0.001, L_2 penalty=1e-4
	Gowalla	Embedding size=64, Batch size=512, Lr=0.001, L_2 penalty=1e-4
	Amazon	Embedding size=128, Batch size=512, Lr=0.001, L_2 penalty=1e-4

In the original papers, NARM, SR-GNN, FGNN, LESSR have used Diginetica; LESSR have processed Gowalla datasets. Thus, for these scenarios, we directly implemented the corresponding settings.

[4]We only choose these three metrics because in next-item prediction, HR is identical to Recall, while MRR is identical to MAP (Mean Average Precision).

5.1.4 Hyper-Parameter Setups

We empirically adopt the optimal hyper-parameter settings for all the methods. For the proposed CGSR,we apply one layer WGAT in *Item Representation Learner*, and use Adam optimizer with the initial learning rate (Lr) 0.001 on Diginetica and Gowalla while 0.003 on Amazon. The representation size of each item d_0 and d are 110 on Diginetica, 60 on Gowalla, and 170 on Amazon. All parameters are initialized using Gaussian distribution with a mean of 0 and a standard deviation of 0.1. The L2 penalty is set to $1e-6$ on Diginetica, Gowalla and $5e-6$ on Amazon. Moreover, the batch size is 20 on Diginetica, 40 on Gowalla, and 100 on Amazon. For baselines, we adopt the optimal settings mentioned in either the original papers for these datasets or the original codes. The settings of baselines are shown in Table 2. Noted that for CGSR and the best baseline, for fair comparison, we run each experiment six times, report the average as the final result in Table 3, and conduct pair-wise t-test to validate the significance of the performance difference.

5.2 Experimental Results

Here, we present results to answer the first three RQs $(1-3)$.

5.2.1 Effectiveness of CGSR over Baseline Methods (RQ1).

To demonstrate the overall performance of CGSR, we compare it with SOTA baseline methods. The comparative results on the three datasets are present in Table 3, where we have some interesting observations as below:

Table 3: Performance of all methods on three datasets in terms of $K = 5, 10, 20$. The best performance is boldfaced, and the runner-up is underlined. We compute the improvements that CGSR achieves relative to the best baseline. Statistical significance of pairwise differences of CGSR vs. the best baseline is determined by a paired t-test (*** for p-value $\leq$.001).

		Traditional			RNN-based		GNN-based				
Datasets	**Metrics**	**POP**	**ItemKNN**	**FPMC**	**GRU4Rec**	**NARM**	**SR-GNN**	**FGNN**	**LESSR**	**CGSR**	**Improv.**
Diginetica	HR@5	0.0040	0.1447	0.1416	0.1576	0.2557	0.2673	0.2640	<u>0.2729</u>	**0.3272**	19.90%***
	HR@10	0.0064	0.2130	0.1769	0.2233	0.3647	0.3774	0.3792	<u>0.3852</u>	**0.4422**	14.80%***
	HR@20	0.0970	0.2897	0.2573	0.3004	0.4889	0.5076	0.5113	<u>0.5147</u>	**0.5601**	8.82%***
	MRR@5	0.0019	0.0807	0.0614	0.0894	0.1419	0.1519	0.1457	<u>0.1561</u>	**0.1862**	19.28%***
	MRR@10	0.0022	0.0897	0.0663	0.0981	0.1581	0.1665	0.1610	<u>0.1708</u>	**0.2014**	17.92%***
	MRR@20	0.0024	0.0965	0.0707	0.1034	0.1647	0.1755	0.1702	<u>0.1799</u>	**0.2096**	16.51%***
	NDCG@5	0.0024	0.0965	0.0625	0.1063	0.1658	0.1804	0.1749	<u>0.1849</u>	**0.2210**	19.52%***
	NDCG@10	0.0032	0.1185	0.0718	0.1274	0.2024	0.2159	0.2121	<u>0.2212</u>	**0.2583**	16.77%***
	NDCG@20	0.0040	0.1379	0.0788	0.1469	0.2315	0.2488	0.2455	<u>0.2538</u>	**0.2881**	13.51%***
Gowalla	HR@5	0.0183	0.2614	0.1869	0.2874	0.3506	0.3557	0.3471	<u>0.3577</u>	**0.3802**	6.29%***
	HR@10	0.0277	0.3248	0.2287	0.3558	0.4272	<u>0.4359</u>	0.4281	0.4340	**0.4612**	5.80%***
	HR@20	0.0500	0.3891	0.2834	0.4326	0.4989	<u>0.5149</u>	0.5080	0.5104	**0.5389**	4.66%***
	MRR@5	0.0090	0.1718	0.0976	0.1863	0.2209	0.2383	0.2212	<u>0.2403</u>	**0.2477**	3.08%*
	MRR@10	0.0102	0.1803	0.1089	0.1954	0.2312	0.2490	0.2321	<u>0.2505</u>	**0.2585**	3.19%***
	MRR@20	0.0118	0.1847	0.1116	0.2006	0.2345	0.2546	0.2376	<u>0.2557</u>	**0.2640**	3.25%***
	NDCG@5	0.0113	0.1941	0.1145	0.2115	0.2533	0.2675	0.2526	<u>0.2695</u>	**0.2808**	4.19%***
	NDCG@10	0.0143	0.2146	0.1239	0.2336	0.2782	0.2935	0.2788	<u>0.2942</u>	**0.3070**	4.35%***
	NDCG@20	0.0200	0.2309	0.1348	0.2530	0.2977	<u>0.3137</u>	0.2990	0.3135	**0.3267**	4.14%***
Amazon	HR@5	0.0046	0.0433	0.0372	0.0418	0.0446	0.0550	0.0464	<u>0.0570</u>	**0.0613**	7.54%***
	HR@10	0.0087	0.0515	0.0497	0.0513	0.0571	0.0684	0.0618	<u>0.0691</u>	**0.0714**	3.33%***
	HR@20	0.0160	0.0591	0.0589	0.0649	0.0718	<u>0.0816</u>	0.0781	0.0813	**0.0848**	3.92%***
	MRR@5	0.0021	0.0288	0.0266	0.0283	0.0300	0.0377	0.0304	<u>0.0396</u>	**0.0448**	13.13%***
	MRR@10	0.0026	0.0299	0.0281	0.0297	0.0317	0.0394	0.0325	<u>0.0412</u>	**0.0462**	12.14%***
	MRR@20	0.0031	0.0304	0.0288	0.0303	0.0327	0.0403	0.0336	<u>0.0420</u>	**0.0471**	12.14%***
	NDCG@5	0.0027	0.0325	0.0295	0.0312	0.0336	0.0420	0.0344	<u>0.0439</u>	**0.0489**	11.39%***
	NDCG@10	0.0040	0.0351	0.0313	0.0339	0.0377	0.0463	0.0394	<u>0.0478</u>	**0.0522**	9.21%***
	NDCG@20	0.0059	0.0370	0.0346	0.0369	0.0414	0.0496	0.0435	<u>0.0509</u>	**0.0556**	9.23%***

To demonstrate the overall performance of CGSR, we compare it with SOTA baseline methods. The comparative results on the three datasets are present in Table 3, where we have some interesting observations as below: (1) The DL-based methods (both RNN-based and GNN-based) generally perform better than traditional methods, demonstrating the capability of DL techniques on processing session data for effective recommendation. Particularly, among the traditional methods, ItemKNN, which is grounded on the similarity between items within a session, performs much better than

POP, and slightly better than first-order MC-based FPMC; (2) Across the DL-based methods, GNN-based methods generally outperform RNN-based methods, which validates the effectiveness of GNN models and graph data structures for SR. For the two RNN-based methods, NARM performs better than GRU4Rec, and its performance is comparable to other GNN-based methods; and (3) The performance of CGSR is significantly better than SOTA GNN-based methods, validating its effectiveness of distinguishing causality relationship between items from correlation relationship. Among all the GNN-based baselines, LESSR performs the best as it captures both local and long-range dependencies among items.

From Table 3, we also observe that the improvements of CGSR on Diginetica and Amazon are both larger than those on Gowalla. This can be partially explained by Figures 2 and 3 which show that stronger causality relationship between items is more prevalent on Diginetica (Amazon) than that on Gowalla. Besides, Gowalla relates to location-based social networking where users share their locations by check-ins. In this case, the directed relationship between check-ins might not so significant compared to that on typical transaction datasets like Diginetica and Amazon.

Table 4: Impact of causality and correlation.

Datasets	Metrics	CGSR-ca	CGSR-r	CGSR-p	CGSR
Diginetica	HR@5	0.3018	0.2674	0.3229	**0.3272**
	HR@10	0.4088	0.3839	0.4380	**0.4422**
	HR@20	0.5232	0.5150	0.5568	**0.5601**
	MRR@5	0.1709	0.1482	0.1804	**0.1862**
	MRR@10	0.1851	0.1636	0.1957	**0.2014**
	MRR@20	0.1932	0.1727	0.2040	**0.2096**
	NDCG@5	0.2008	0.1776	0.2157	**0.2210**
	NDCG@10	0.2354	0.2153	0.2529	**0.2583**
	NDCG@20	0.2643	0.2483	0.2830	**0.2881**
Gowalla	HR@5	0.3718	0.3601	0.3798	**0.3802**
	HR@10	0.4516	0.4388	0.4601	**0.4612**
	HR@20	0.5272	0.5183	0.5386	**0.5389**
	MRR@5	0.2399	0.2338	0.2467	**0.2477**
	MRR@10	0.2451	0.2444	0.2575	**0.2585**
	MRR@20	0.2532	0.2499	0.2629	**0.2640**
	NDCG@5	0.2708	0.2653	0.2802	**0.2808**
	NDCG@10	0.2954	0.2908	0.3063	**0.3070**
	NDCG@20	0.3143	0.3109	0.3258	**0.3267**
Amazon	HR@5	0.0567	0.0530	0.0555	**0.0613**
	HR@10	0.0691	0.0648	0.0681	**0.0714**
	HR@20	0.0830	0.0784	0.0820	**0.0848**
	MRR@5	0.0385	0.0373	0.0401	**0.0448**
	MRR@10	0.0409	0.0389	0.0418	**0.0462**
	MRR@20	0.0418	0.0400	0.0428	**0.0471**
	NDCG@5	0.0438	0.0412	0.0440	**0.0489**
	NDCG@10	0.0480	0.0450	0.0480	**0.0522**
	NDCG@20	0.0516	0.0485	0.0515	**0.0556**

5.2.2 Effectiveness of Causality Graphs vs. Correlation Graph (RQ2).

CGSR considers both causality and correlation graphs. To explore the effectiveness of each type of relationship, we compare CGSR with three variants: (1) CGSR-ca removes both cause graph $\mathcal{G}^c$ and effect graph $\mathcal{G}^e$; (2) CGSR-r ignores correlation graph $\mathcal{G}^r$; (3) CGSR-p does not consider the session representation $\mathbf{S}^p$. The performance of CGSR and the three variants are shown in Table 4.

As shown in Table 4, CGSR performs superior to the three variants across all metrics, validating the effectiveness of the three designs, particularly distinguishing directed causality relationship from undirected correlation relationship between items. Besides, CGSR-ca performs better than CGSR-r, implying that only considering correlation graph is better than only considering causality relationship. This might be due to the undirected correlation graph might already cover the causality relationship but vice versa not.

Furthermore, we also explore the effectiveness of weights designs in cause and effect graphs (i.e., **impact of causality weights**). CGSR defines weights of causality-related graphs as discussed in Section 3. To validate the effectiveness, we compare CGSR with two alternatives: (1) CGSR-W, where edge weights in cause and effect graph are set to 1; and (2) CGSR-CC, which does not rule out the impact of common cause. As can see in Figure 8, CGSR performs better than

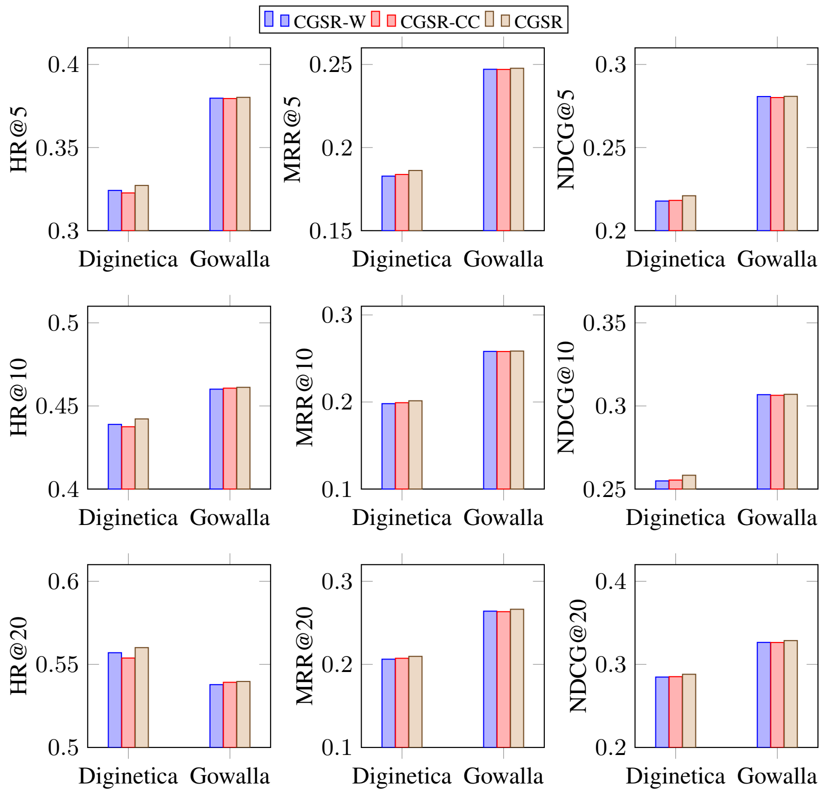


Figure 8: Performance of different causality weights.

the two variants, verifying the effectiveness of our design and the necessity of removing common cause in identifying causality relationship.

5.2.3 Ablation Study (RQ2).

Besides causality graphs, we have some other innovative designs in CGSR: (1) three types of second-order relationship in building correlation graph; (2) first-order relationship in building cause and effect graphs; and (3) adopting one layer WGAT in Item Representation Learner.

Towards the first issue, to explore **the effectiveness of the second-order relationship**, we compare our model with three variants: (1) CGSR-chain removes the "chain" relationship; (2) CGSR-fork ignores "fork" type; and (3) CGSR-collider does not consider second-order neighbors of collider type. Table 5 summarizes the comparative results and shows that all the three types contribute to performance improvement, but fork relation is less significant than the other two. This is consistent with directed graphical model principles which indicate that v_i and v_j in Figure 5 incline to be independent with each other given a known v_k in fork pattern [47].

Towards the second issue, we compare CGSR with the variant: CGSR-ca_sec, which considers the second-order relationship in cause and effect graphs. The results of CGSR and the variant are shown in Table 6. From Table 6, CGSR performs superior to the variant across all metrics, partially validating the effectiveness of only considering first-order relationship in our study.

Towards the third issue, other models like gated graph neural network (GGNN) [13] are also suitable for directed and weighted graphs. To verify the validity of one layer WGAT for CGSR, we consider model variants by varying the number of GGNN layers and WGAT layers, respectively. The comparative results regarding these variants are summarized in Table 7, which shows that WGAT-related variants performs better than GGNN-related variants, and one layer WGAT consistently performs better across all scenarios, demonstrating the effectiveness of our design in *Item Representation Learner*.

Table 5: Impact of second-order relationship in correlation graph.

Datasets	Metrics	CGSR-chain	CGSR-fork	CGSR-collider	CGSR
Diginetica	HR@5	0.3205	0.3211	0.3168	**0.3272**
	HR@10	0.4340	0.4378	0.4343	**0.4422**
	HR@20	0.5528	0.5557	0.5534	**0.5601**
	MRR@5	0.1803	0.1834	0.1789	**0.1862**
	MRR@10	0.1955	0.1990	0.1946	**0.2014**
	MRR@20	0.2038	0.2072	0.2028	**0.2096**
	NDCG@5	0.2151	0.2175	0.2131	**0.2210**
	NDCG@10	0.2518	0.2553	0.2511	**0.2583**
	NDCG@20	0.2819	0.2851	0.2812	**0.2881**
Gowalla	HR@5	0.3754	0.3780	0.3759	**0.3802**
	HR@10	0.4552	0.4596	0.4560	**0.4612**
	HR@20	0.5327	0.5372	0.5335	**0.5389**
	MRR@5	0.2457	0.2459	0.2450	**0.2477**
	MRR@10	0.2564	0.2569	0.2557	**0.2585**
	MRR@20	0.2617	0.2623	0.2611	**0.2640**
	NDCG@5	0.2781	0.2788	0.2776	**0.2808**
	NDCG@10	0.3039	0.3053	0.3036	**0.3070**
	NDCG@20	0.3235	0.3249	0.3232	**0.3267**

Table 6: Impact of first-order relationship in cause and effect graphs.

Datasets	Metrics	CGSR-ca_sec	CGSR
Diginetica	HR@5	0.3190	**0.3272**
	HR@10	0.4367	**0.4422**
	HR@20	0.5569	**0.5601**
	MRR@5	0.1802	**0.1862**
	MRR@10	0.1959	**0.2014**
	MRR@20	0.2043	**0.2096**
	NDCG@5	0.2146	**0.2210**
	NDCG@10	0.2527	**0.2583**
	NDCG@20	0.2831	**0.2881**
Gowalla	HR@5	0.3624	**0.3802**
	HR@10	0.4448	**0.4612**
	HR@20	0.5254	**0.5389**
	MRR@5	0.2273	**0.2477**
	MRR@10	0.2384	**0.2585**
	MRR@20	0.2440	**0.2640**
	NDCG@5	0.2610	**0.2808**
	NDCG@10	0.2878	**0.3070**
	NDCG@20	0.3081	**0.3267**

5.2.4 Sensitivity of Hyper-parameters (RQ3).

We investigate the impact of embedding size d and batch size on CGSR model, by deploying a grid search in the range of $\{40, 50, 60, 70, 80, 90, 100, 110, 120\}$ and $\{20, 40, 60, 80, 100\}$ for d and batch size, respectively. Figure 9 shows the experiment results. Generally speaking, our method is comparatively insensitive to the two hyper-parameters. Besides, while varying the hyper-parameters, the performance on Diginetica fluctuates more obviously, which is consistent with our previous analysis that causality patterns are more prevalent on Diginetica than on Gowalla.

5.3 Case Study on Explanation Task (RQ4)

Since CGSR has distinguished causality from correlation relationship between items, it is expected to facilitate the explanation task in SR in a fine-grained fashion. In this case, we design an explainable framework on CGSR to clarify why a specific item $v_j \in \mathcal{V}$ is recommended given session S on both session and item levels by generating a set of explanation scores. Specifically, in the *explainable framework*, **on the session level**, three scores leading to the final recommendation score of v_j (i.e., $Score_j^{ca}, Score_j^{r}, Score_j^{p}$) are output by *Recommendation Score Generator* in CGSR. **On the item level**, we calculate the score (importance) of each item v_i^S in S to v_j under each relationship type (i.e.,

Table 7: Performance with different GNN layers.

Datasets	Metrics	3*GGNN	2*GGNN	GGNN	3*WGAT	2*WGAT	WGAT+GGNN	WGAT
Diginetica	HR@5	0.2794	0.2787	0.2794	0.2836	0.2952	0.3101	**0.3272**
	HR@10	0.3934	0.3921	0.3934	0.4038	0.4159	0.4268	**0.4422**
	HR@20	0.5198	0.5197	0.5198	0.5310	0.5427	0.5473	**0.5601**
	MRR@5	0.1598	0.1587	0.1598	0.1571	0.1625	0.1722	**0.1862**
	MRR@10	0.1750	0.1738	0.1750	0.1731	0.1785	0.1877	**0.2014**
	MRR@20	0.1837	0.1836	0.1837	0.1819	0.1873	0.1961	**0.2096**
	NDCG@5	0.1894	0.1884	0.1894	0.1884	0.1953	0.2064	**0.2210**
	NDCG@10	0.2262	0.2250	0.2262	0.2272	0.2343	0.2441	**0.2583**
	NDCG@20	0.2582	0.2572	0.2582	0.2594	0.2663	0.2745	**0.2881**
Gowalla	HR@5	0.3570	0.3652	0.3471	0.3489	0.3613	0.3730	**0.3802**
	HR@10	0.4370	0.4438	0.4270	0.4276	0.4425	0.4547	**0.4612**
	HR@20	0.5167	0.5219	0.5081	0.5056	0.5213	0.5349	**0.5389**
	MRR@5	0.2322	0.2392	0.2270	0.2209	0.2311	0.2432	**0.2477**
	MRR@10	0.2429	0.2497	0.2377	0.2315	0.2420	0.2541	**0.2585**
	MRR@20	0.2484	0.2551	0.2434	0.2369	0.2474	0.2597	**0.2640**
	NDCG@5	0.2633	0.2706	0.2569	0.2529	0.2635	0.2756	**0.2808**
	NDCG@10	0.2892	0.2860	0.2828	0.2783	0.2899	0.3020	**0.3070**
	NDCG@20	0.3093	0.3158	0.3043	0.2981	0.3098	0.3223	**0.3267**

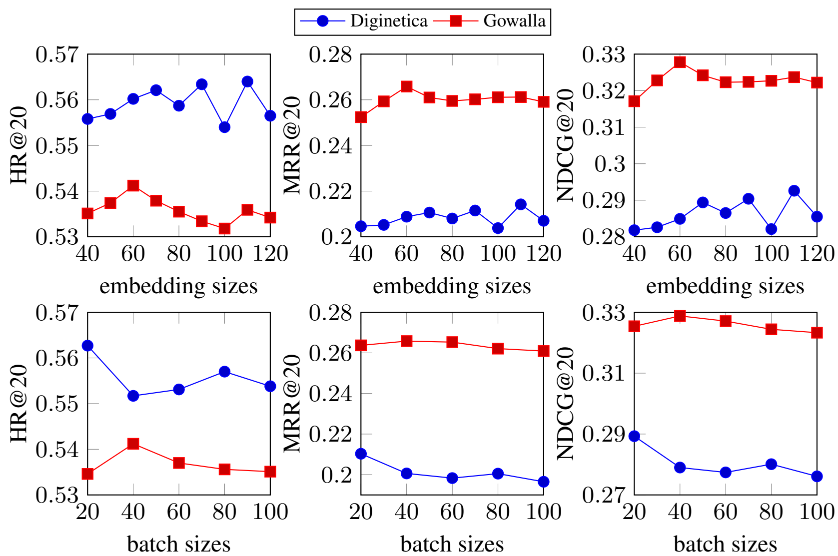


Figure 9: Model performance of different embedding sizes and batch sizes ($K = 20$).

$Score^{ca}_{ij}$ for causality relationship and $Score^{r}_{ij}$ for correlation relationship):

$$
\begin{aligned}
Score^{ca}_{ij} &= (\mathbf{W_{c,6}}[\mathbf{x}^{c}_{i,S};\mathbf{x}^{c}_{i,S}])^{T}\mathbf{x}^{e}_{j} - \gamma_1(\mathbf{W_{e,6}}[\mathbf{x}^{e}_{i,S};\mathbf{x}^{e}_{i,S}])^{T})\mathbf{x}^{c}_{j} \\
Score^{r}_{ij} &= (\mathbf{W_{r,6}}[\mathbf{x}^{r}_{i,S};\mathbf{x}^{r}_{i,S}])^{T}\mathbf{x}^{r}_{j}
\end{aligned}
\tag{18}
$$

Consequently, with the framework, given a recommended item v_j and session S, we can not only understand the impact of S under each type of relationship on v_j from the session level, but also recognize the importance of each item in S as either cause item or correlation item to v_j from the item level.

In our experiment, to showcase the effectiveness of our explainable framework, we instantiate it on Amazon dataset. We choose Amazon dataset because item information on Amazon is publicly available while on other two datasets it is anonymously encoded. Figure 10 depicts two cases on Amazon, which relate to two randomly chosen sessions: S_1 {Cake Lifter, Cooling Rack, Griddler, Griddler Waffle Plates (recommended item)} and S_2 {Mini-Prep Plus Food Processor, Oven with Dual Handles, Slow Cooker, Griddler (recommended item)}. The histogram on the left side presents the explainable scores on session level, whilst the histogram on the right side shows those on item level.

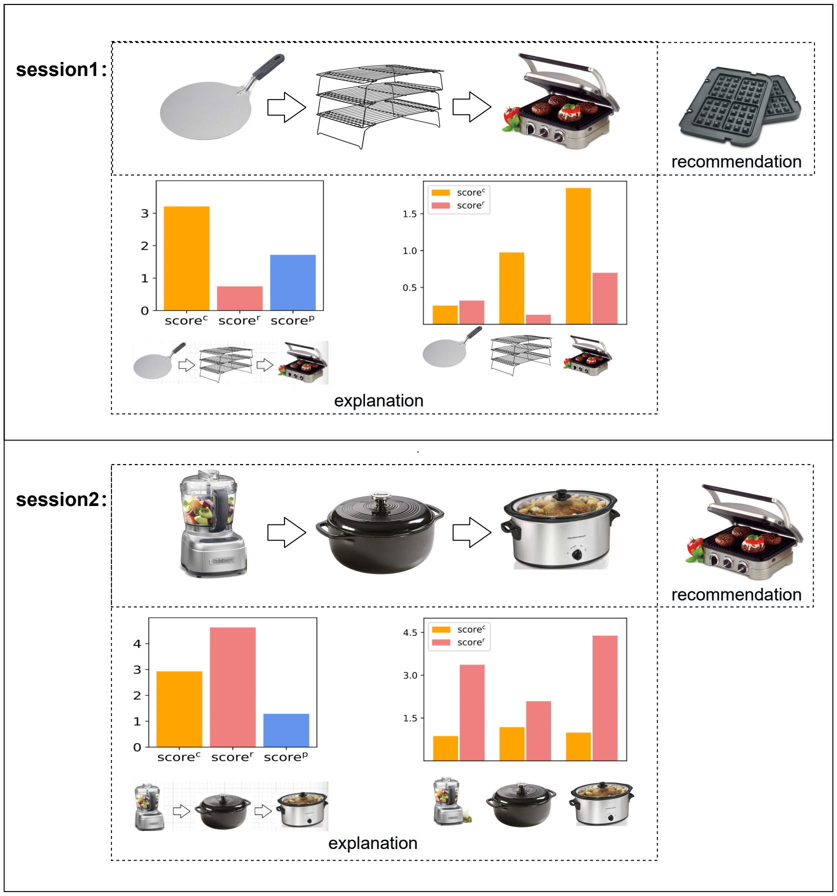


Figure 10: Two sessions {Cake Lifter, Cooling Rack, Griddler, Griddler Waffle Plates} and {Mini-Prep Plus Food Processor, Oven with Dual Handles, Slow Cooker, Griddler}, and the corresponding explanation scores on both session level and item level.

As shown in Figure 10, for S_1, causality score $Score^{ca}$ is higher than correlation score $Score^{r}$, implying that on session level, Cake Lifter, Griddler and Cooling Rack in S_1 are more like cause items leading to buy Griddler Waffle Plates. On item level, under causality relationship, the score of Griddler is higher than the other two items, which means that Griddler is the main reason motivating to also purchase the Griddler Waffle Plates. Towards S_2, correlation score is higher than causality score. For each item in S_2, the corresponding correlation score with Griddler is also higher than the causality score. This is consistent with reality that Griddler has a correlation relationship rather than a causality relationship with Mini-Prep Plus Food Processor, Oven with Dual Handles, and Slow Cooker, respectively. From the two case studies, we can see that our explainable framework can provide valuable and reasonable explanations towards recommendations in SR.

6 Conclusion

The directed causality relationship between items, which has been ignored by previous studies, is quite important for effective session-based recommendation (SR). In this paper, we proposed a novel method denoted as CGSR to explicitly consider causality and correlation relationship in SR. Specifically, on the basis of sessions, we constructed a cause graph, an effect graph, and a correlation graph considering both first-order and three types of second-order relationship, which are fed into a GNN and attention mechanism-based model to obtain four types of session representations for recommendation. Experimental results on three real-world datasets demonstrated the superiority of model over the state-of-the-art, and validated the effectiveness of every component in CGSR. We further designed an explainable framework on CGSR to improve the explainability of SR. Case studies on Amazon dataset showcased that our framework can facilitate the explanation task in SR.

7 Acknowledgments

We greatly acknowledge the support of National Natural Science Foundation of China (Grant No. 72192832), and the Natural Science Foundation of Shanghai (Grant No. 21ZR1421900).

References

[1] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems*, 39(1):1–42, 2020.

[2] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 811–820, 2010.

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations*, 2016.

[4] Balázs Hidasi and Alexandros Karatzoglou. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 843–852, 2018.

[5] Xiangde Zhang, Yuan Zhou, Jianping Wang, and Xiaojun Lu. Personal interest attention graph neural networks for session-based recommendation. *Entropy*, 23(11):1500, 2021.

[6] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *IEEE International Conference on Data Mining*, pages 197–206. IEEE, 2018.

[7] Feng Yu, Yanqiao Zhu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. Tagnn: Target attentive graph neural networks for session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1921–1924, 2020.

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

[9] Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference. *arXiv preprint arXiv:2002.02770*, 2020.

[10] Bernhard Schölkopf. Causality for machine learning. *arXiv preprint arXiv:1911.10500*, 2019.

[11] Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, pages 507–517, 2016.

[12] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pages 346–353, 2019.

[13] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

[14] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin. 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*, pages 579–588, 2019.

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

[16] Xin Liu, Yong Liu, Karl Aberer, and Chunyan Miao. Personalized point-of-interest recommendation by mining users' preference transition. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 733–738, 2013.

[17] Duc-Trong Le, Yuan Fang, and Hady W Lauw. Modeling sequential preferences with dynamic user and context factors. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–161. Springer, 2016.

[18] Xiang Wu, Qi Liu, Enhong Chen, Liang He, Jingsong Lv, Can Cao, and Guoping Hu. Personalized next-song recommendation in online karaokes. In *Proceedings of the 7th ACM Conference on Recommender Systems*, pages 137–140, 2013.

[19] Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on Deep Learning for Recommender Systems*, pages 17–22, 2016.

[20] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1441–1450, 2019.

[21] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 729–732, 2016.

[22] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 130–137, 2017.

[23] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. Sequential user-based recurrent neural network recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems*, pages 152–160, 2017.

[24] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 26th ACM International Conference on Conference on Information and Knowledge Management*, pages 1419–1428, 2017.

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

[26] Ruihong Qiu, Zi Huang, Jingjing Li, and Hongzhi Yin. Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Transactions on Information Systems*, 38(3):1–23, 2020.

[27] Ruihong Qiu, Hongzhi Yin, Zi Huang, and Tong Chen. Gag: Global attributed graph neural network for streaming session-based recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 669–678, 2020.

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

[29] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou. Graph contextualized self-attention network for session-based recommendation. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, volume 19, pages 3940–3946, 2019.

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

[31] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 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*, pages 169–178, 2020.

[32] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 43–52, 2015.

[33] Yin Zhang, Haokai Lu, Wei Niu, and James Caverlee. Quality-aware neural complementary item recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 77–85, 2018.

[34] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2015.

[35] Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommendation. In *Causation: Foundation to Application, Workshop at UAI*. AUAI, 2016.

[36] Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims. Recommendations as treatments: Debiasing learning and evaluation. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1670–1679. PMLR, 2016.

[37] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. The deconfounded recommender: A causal inference approach to recommendation. *arXiv preprint arXiv:1808.06581*, 2018.

[38] Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. Doubly robust joint learning for recommendation on data missing not at random. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6638–6647. PMLR, 2019.

[39] Stephen Bonner and Flavian Vasile. Causal embeddings for recommendation. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 104–112, 2018.

[40] Yixin Wang, Dawen Liang, Laurent Charlin, and David M Blei. Causal inference for recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 426–431, 2020.

[41] Ruihong Qiu, Sen Wang, Zhi Chen, Hongzhi Yin, and Zi Huang. Causalrec: Causal inference for visual debiasing in visually-aware recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3844–3852, 2021.

[42] Ben Taskar, Ming-Fai Wong, Pieter Abbeel, and Daphne Koller. Link prediction in relational data. *Advances in Neural Information Processing Systems*, 16:659–666, 2003.

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

[44] Lei Guo, Hongzhi Yin, Qinyong Wang, Tong Chen, Alexander Zhou, and Nguyen Quoc Viet Hung. Streaming session-based recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1569–1577, 2019.

[45] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*, pages 565–573, 2018.

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

[47] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.