Spatio-Temporal Contrastive Learning Enhanced GNNs for Session-based Recommendation

Zhongwei Wan, Benyou Wang, Xin Liu, Jiezhong Qiu, and Guangyong Chen*,

Abstract—Session-based recommendation (SBR) systems aim to utilize the user's short-term behavior sequence to predict the next item without the detailed user profile. Most recent works try to model the user preference by treating the sessions as between-item transition graphs and utilize various graph neural networks (GNNs) to encode the representations of pair-wise relations among items and their neighbors. Some of the existing GNN-based recommendation models usually suffer from temporal information loss and they mainly focus on aggregating information from the view of spatial structure information, which ignore the temporal relations within neighbors during message passing and result in sub-optimal problem. Another works embrace this challenge by incorporating additional temporal information, but lack sufficient interaction between the spatial and temporal patterns. To address this issue, inspired by the uniformity property of contrastive learning techniques, we propose a novel framework called Session-based **RE**commendation with **S**patio-**T**emporal **C**ontrastive Learning Enhanced GNNs (**RESTC**). The idea is to supplement the GNN-based main supervised recommendation task with the temporal representation via an auxiliary cross-view contrastive learning mechanism. In particular, RESTC attempts to remedy the temporal information loss and refine the session representations by encoding both temporal and spatial structure patterns of item transitions with two distinct encoders, where it aligns the embeddings of the two views in a unified latent space via self-discrimination . Furthermore, a novel global collaborative filtering graph (CFG) embedding is leveraged to enhance the spatial view in the main task. Extensive experiments demonstrate the significant performance of RESTC compared with the state-of-the-art baselines e.g., with an improvement as much as 27.08% gain on HR@20 and 20.10% gain on MRR@20.

Index Terms—Recommendation system; Session-based recommendation; Graph neural network; Temporal; Contrastive learning

1 Introduction

Recommendation systems have been the efficient tool for helping users make informative choices according to their available profiles and the preferences reflected in the long term history interaction, which are widely used in web search and various stream medias [1], [2], [3]. However, the traditional recommenders may perform poorly in some scenarios where the user's interaction is short in a narrow period or the status is unlogged-in. Thus, Session-based Recommendation (SBR) has attracted increasing research [4], [5], [6], [7], since it characterizes users' short-term preference from the limited interactions in the current session, e.g., a basket of products purchased in one transaction visit, and then predict the products that a user interacts with in the future.

Conventional SBR can be mainly divided into two paradigms. One key line of research treats each session as a sequence of items sorted by interactive time and utilizes Recurrent Neural Networks (RNN) [8], [9], [10] or memory networks [11] to learn sequential behaviors in a local session to represent the preference and interests of users. Although effective performance has been achieved, modeling sequences directly are arguably insufficient to obtain accurate user representation in sessions and neglect complex transition patterns of items [5]. Instead, the other paradigm constructs graph structure from the session and leverages Graph Neural Networks (GNNs) [5], [6], [12], [13], [14] to conduct information aggregation between adjacent items and capture complex high-order relations, which obtains significant superiority over sequence-based methods.

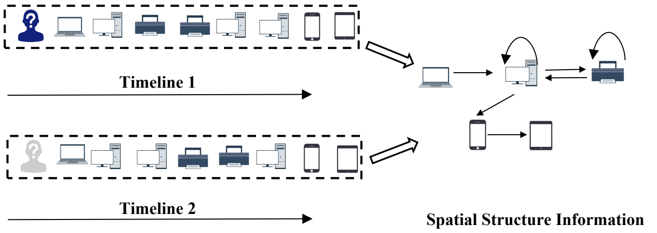
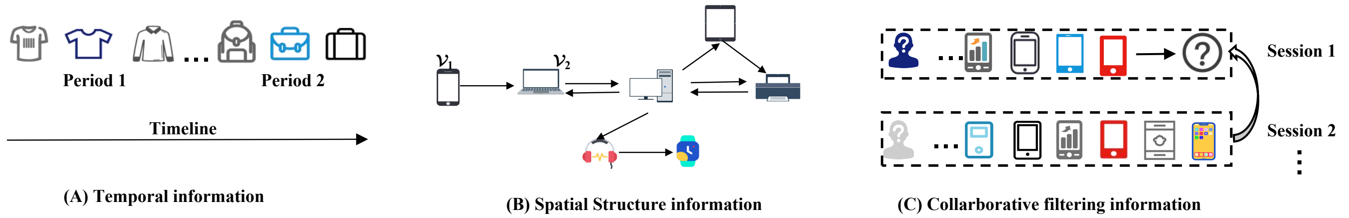


Fig. 1: Two distinct sessions may be represented as the same graph if the temporal information is omitted, indicating the temporal pattern should be sufficiently considered to supplement GNN-based models for SBR task.

However, the temporal information has been omitted by the abovementioned GNN-based methods because of the permutation-invariant aggregation during the message passing in the spatial structure, which is a vital signal that contributes significantly to capture the preference evolution of the user in the temporal dimension [15], [16]. A concrete example for the impact of the temporal information loss is shown in Figure 1. For instance, two distinct anonymous electronic-product-shopping sessions may be represented as the same graph. In fact, if the two sessions produce the different next item but they are encoded as the same graph representation since the aggregation function of GNN could not distinguish the temporal order of items' neighbors, it

- *Zhongwei Wan is with the Department of Artificial Intelligence, University of Chinese Academy of Sciences and Shenzhen Institute of Advanced Technology (CAS). Shenzhen, China.*
  E-mail:zw.wan1@siat.ac.cn
- **Corresponding author*

Manuscript received April 19, 2005; revised August 26, 2015.



(A) Temporal information

(B) Spatial Structure information

(C) Collarborative filtering information

Fig. 2: Three essential information among sessions data: (A) *temporal view of a session* is about a behavioral sequence containing user's dynamic preference w.r.t its timeline; (B) *spatial view of a session* refers to a between-item transition directed graph, each edge of which indicates a behavior shift from the source item to the target item — for example, a user has clicked item v_2 after v_1. Note that behavior shift associated with an edge could happen many times in a session, and such edges are orthogonal to time; (C) *collaborative filtering information in other sessions* could be extracted from a global weighted graph then used to compensate for the item profiles in a short-term session.

will make the GNN based model induce incorrect results and limit the its capacity without the essential temporal pattern. Fortunately, some line of works have attempted to incorporate temporal information by modeling a session as the dynamic sub-session graphs at the fixed-length time intervals [17], [18] or integrating the timestamps information as a contextual dimension [19]. However, modeling multiple sub-session graphs based on timeline may introduce redundant spatial structure information and it still miss temporal order during aggregation. Besides, all of these methods lack sufficient interaction between spatial structure and temporal patterns in the latent space, which restricts the representation capability of the models.

Therefore, incorporating temporal pattern then modeling the latent mutual presentation of spatial and temporal views of a session is crucial and challenged for session-based recommendation systems. To align the embeddings of the two views in a unified latent space, (i) one straightforward way could be to directly adopt concatenation or cross-attention based methods [20], [21] to fuse these two information resources after the encoding phase. But both views know little information about each other in this way since there is no efficient interaction between two different encoders during training. (ii) The other approaches could be to utilize Coupled GAN [22] to learn the joint distribution of multi-style views or leverage semi-supervised learning paradigm like Co-training [23] to acquire complementary information from each other view. However, it is unstable to optimize the min-max objective of GAN-style methods. Besides, both GAN and Co-training mechanisms face the mode collapse problem [24] while learning the latent representation of different views during training the models.

Due to the issues mentioned above, inspired by the uniformity property [25] and theoretical guarantee for semantic representation alignment in latent space [26] of contrastive learning, we propose a novel auxiliary **spatio-temporal contrastive learning** framework named RESTC, which can align the spatial and temporal semantic representations in a projected feature space to conserve as much mutual information of the two views as possible and is easier to optimize. Although existing contrastive learning techniques for sequential [27], [28], [29] or GNN-based recommendation [30], [31], [32] generally generate positive samples using item-level augmentation,e.g., item cropping, masking, reordering or sub-sample in sequence and graph data, which are not suitable for SBR since these methods induce semantically inconsistent samples and damage the completeness of temporal pattern. Different to the above works, we comprehensively consider two different views on session-level, which adopts a spatial encoder for the graph structure representation learning and a temporal encoder to supplement the temporal representation as the informative positive sample.

Specifically, it is worthwhile to notice that our RESTC is model-agnostic that can be applied to any GNN-based models, here we employ the powerful Multi-relational Graph Attention Network (MGAT) [33], [34] as the spatial encoder. We further derive a well-designed Session Transformer (SESTrans) contained a temporal enhanced module as the temporal encoder. For the contrastive objective, we propose a mixed noise negative sampling strategy different from [35] to further enhance the model performance. With the contrastive learning loss, we enhance the cross-view interaction in the latent space to refine session representation by maximizing the agreement of positive pairs. Furthermore, due to the data sparsity of short-term session data, a Collaborative Filtering Graph (CFG) derived from all sessions as a global weighted item transition graph, is leveraged to enhance the spatial view with the collaborative filtering embedding in the main supervised task. The example of spatial, temporal and collaborative information of a session are shown in Figure 2. The experiment results show that RESTC outperforms the state-of-the-art baselines, showing the effectiveness of incorporating temporal information with spatio-temporal contrastive learning. Our **contributions** can be summarized as below.

- We highlight the significance of incorporating temporal information for GNN-based SBR task, facilitating the development of cross-view interaction for the spatial and temporal pattern.
- To best of our knowledge, the proposed spatio-temporal contrastive learning framework RESTC is the first work aiming to align and refine the representations of spatial and temporal views in the latent space, which can adaptively plugged into any existing GNN-based models.
- We conduct extensive experiments on six real-life public datasets, demonstrating that our model consistently outperforms the state-of-the-art methods with a large margin.

2 Related Work

2.1 Sequence-based Models in SBR

In early research, FPMC [36] utilized Markov chain and matrix factorization to obtain the sequential pattern of session. Recently, neural network-based models have demonstrated effectiveness in exploiting sequential data in SBR tasks. GRU4Rec [10] was the first RNN-based model which captured item transitions by multi-layer GRUs. NARM [4] leveraged an attention-based method to combine RNN to model complex items relations better. STAMP [11] used the attention-based memory network to capture the user's current interest. Inspired by Transformer architecture, SASRec [37] stacked several self-attention layers to model the item-transition sequence. BERT4Rec [38] employed deep bidirectional self-attention to model user behaviors for sequence recommendation. Besides, Yuan et al. [39] also propose to use a dual sparse attention network to explore the current user's interest via an adaptively learnable target embedding. These attention-based models separately deal with the user's last item and the whole current session, thus capturing the user's general and recent interest. However, modeling the session as a sequence directly is hard to obtain complex transition patterns of items [5].

2.2 GNN-based Models in SBR

Most recent works focus on utilize Graph Neural Networks (GNNs) to extract the relationship in the session, which have shown better results than sequence-based models [5], [6], [12]. For instance, SR-GNN [5] used a gate GNN model to obtain item embeddings over an item graph and predict the next item using the attention mechanism. GC-SAN [6] utilized self-attention networks to aggregate the information of session graphs. FGNN [12] leveraged multi-head attention to aggregate the neighbor item's embeddings in a weighted item-transition graph. LESSR [14] preserved session order based on GRU and shortcut graph attention to solve the lossy session encoding and ineffective long-range dependency capturing problems. Zhou and Pan et al. [17], [18] constructed a sequence of dynamic graph snapshots at timestamps to model the preference evolution. GCE-GNN [40] proposed to exploit a session-graph convolution and global neighbor graph convolution to conduct a more accurate session embedding. TMI-GNN [19] proposed to use temporal information to guide the multi-interest network to focus on multi-interest mining. Although some GNN-based methods have attempted to incorporate temporal information, these works model spatial structural and temporal patterns separately without taking account into their interactions in the latent space, which restricts their representation ability.

2.3 Contrastive Learning in RS

Recently, In the CV and NLP area, multiple contrastive Learning [35], [41], [42], [43] methods have demonstrated superior performance in modeling representation by measuring the similarity between different views within unlabeled raw data. This self-supervised mechanism is widely adopted in recommendation systems because it carries good semantic or structural meanings and benefits downstream tasks. For instance, GCC [44] proposed sub-graph instance discrimination that utilized contrastive learning to learn the intrinsic and transferable structural representations. Yao et al. [45] proposed multi-task contrastive learning for a two-tower model. Besides, S^3-Rec [46] made use of the mutual information maximization to explore the correlation among items, attributes, and contexts. Recently, Wei et al. proposed CLCRec [47] to leverage contrastive learning to learn the mutual dependencies between item content and collaborative signals in order to solve the cold start problem. Wu et al. [32] generated multiple views of the same node from a graph and employed contrastive learning to maximize their agreement to mine hard negative samples. In SBR task, Li et al. [48] made use of a global-level contrastive learning model to solve noise and sampling problems in heterogeneous graphs. S^2-DHCN [49] is the most relevant work to us, which designs a contrastive learning mechanism to enhance hyper-graph modeling via another line GCN model. But it still suffers from the temporal information loss in the spatial structure, leading to sub-optimal performance. Orthogonal these methods, our RESTC employs spatio-temporal contrastive learning to supply sufficient interaction between spatial structure and temporal pattern via aligning the two views in the latent space.

3 Problem Definition

Suppose that the item set is $V = \{v_1, v_2, \ldots, v_{|N|}\}$, where v_i indicates item i and and $|N|$ is the number of item categories. Given an ongoing session denoted as items $s = [v_1, v_2, v_3, \ldots, v_M]$, $v_i \in V(1 \leq i \leq |N|)$ represents the i-th historical interactive item of the user within session s, and M is the length of the session, it aims to predict the items v_{M+1} that the user will interact with at the next time stamp. Generally, the goal of the session-based recommendation is to recommend the top-K rank items $(1 \leq K \leq |N|)$ that have the highest probabilities to be clicked/purchased by the user.

4 Spatio-Temporal Contrastive Learning

In this section, we augment a session into two views of embeddings from a **temporal encoder** in Sec. 4.2 and a **spatial encoder** 4.1 respectively. In order to align and interact the output embeddings from the two encoders in the latent space, we design a contrastive learning task and introduce it in Sec. 4.3.

4.1 Temporal Encoder for Session Sequences

We here present how to model session data as sequences from a temporal view, corresponding to the temporal part of Fig. 3.

4.1.1 *Session Sequence Construction*

Given a session $s = [v_1, v_2, v_3, \ldots, v_M]$, by adopting an embedding layer, all items in the session will be embedded to a sequence of item embeddings, denoted as $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3} \ldots, \mathbf{x_L}]$, $\mathbf{X} \in \mathbb{R}^{L \times D}$ is the model input. L denotes the max length of all sessions; the zero vector will be padded after the sequence when the length of a session M is shorter than L. To aggregate item embeddings

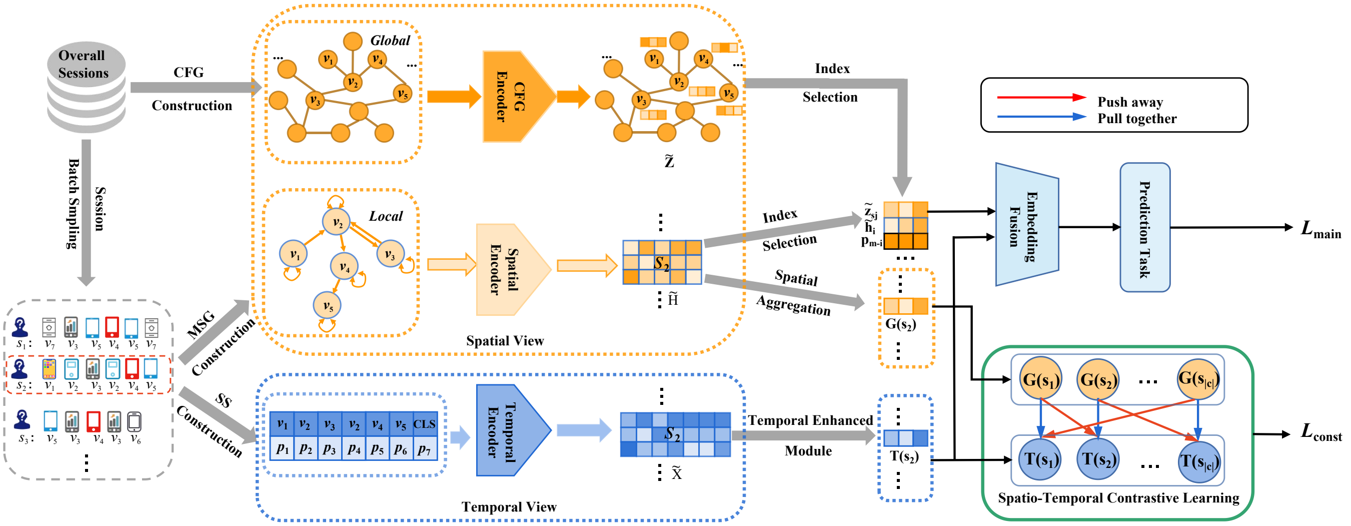


Fig. 3: Overview of our proposed RESTC.

to a fused session representation as a temporal pattern, we add a special item [CLS] at the end of the session sequence, which is similar to BERT [50]. To encode temporal information, we equip the intial item embeddings with the learnable absolute temporal position embeddings (denoted as $\mathbf{P_t} \in \mathbb{R}^{L\times D}$):

$$\mathbf{X}' = \text{Concat}(\mathbf{X_t}, \mathbf{P_t}), \tag{1}$$

where $\mathbf{X}' \in \mathbb{R}^{(L+1)\times 2D}$.

4.1.2 Session Transformer Layers for SEs

To obtain preliminary temporal embedding of sessions , we leverage the Self-Attention Network (SAN) following the standard transformer encoder [51], which employs weight matrix $\mathbf{W}_Q$, $\mathbf{W}_K$, $\mathbf{W}_V$ to linearly transform the input $\mathbf{X}' \in \mathbb{R}^{(L+1)\times 2D}$ as query, key, value vectors, denoted as $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$. The scaled dot-product attention is defined as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{2D}}\right)\mathbf{V}. \tag{2}$$

Intuitively, the attention module aggregates low-level item representations to high-level item representations via a linear combination. We also implement SAN in a multi-head fashion like in [51]. Since SAN is linear to input, we feed the output of SAN to a feed-forward network (FFN) with non-linearity activation:

$$\text{FFN}(\mathbf{x}) = \text{ReLU}\left(\mathbf{x}\mathbf{W_1} + \mathbf{b_1}\right)\mathbf{W_2} + \mathbf{b_2}, \tag{3}$$

where $\mathbf{W_1}$ and $\mathbf{W_2} \in \mathbb{R}^{2D\times 2D}$ and $\mathbf{b_1}$, $\mathbf{b_2} \in \mathbb{R}^{2D}$ are trainable parameters in FFN layers. Besides, we stack several encoder layers to learn more complicated session representation from the temporal review, accompanied by standard residual connection, dropout mechanism, and layer normalization. After that, we obtain the encoder's output embedding $\tilde{\mathbf{X}}$.

4.1.3 Temporal Enhanced Module

To better aggregate item embeddings from encoder layers to obtain the user's evolving preference with respect to the timeline, we develop a novel temporal enhanced module. In particular, we utilized the embedding of the special item [CLS] of output embeddings $\tilde{\mathbf{X}}$ as query vector $\mathbf{Q}'$, and the rest of output embeddings $\tilde{\mathbf{X}}$ as key vector $\mathbf{K}'$. Note that $\mathbf{Q}'$ is the global preference representation, and $\mathbf{K}'$ is the preference evolution representation. Besides, we leverage initial embedding $\mathbf{X}'$ as our value vector $\mathbf{V}'$ since it contains the original temporal positional encoding information, which can benefit our output embedding with the temporal pattern. Then, we add the two representations and apply a non-linear transformation with ReLU activation. Finally, a softmax function is used to calculate attentive relations and gain the aggregative vector $\mathbf{h_t}$. The formulas are defined as:

$$\gamma_\mathbf{t} = \text{softmax}\left(\text{ReLU}\left(\mathbf{Q}'\mathbf{W_3} + \mathbf{K}'\mathbf{W_4} + \mathbf{b_3}\right)\mathbf{f_t}\right), \tag{4}$$

$$\mathbf{h_t} = \sum_{i=1}^{L} \gamma_{\mathbf{t}_\mathbf{i}}\, \mathbf{v}'_\mathbf{i}, \tag{5}$$

where $\mathbf{W_3}, \mathbf{W_4} \in \mathbb{R}^{2D\times 2D}$ and $\mathbf{f_t}, \mathbf{b_3} \in \mathbb{R}^{2D}$ $\gamma_\mathbf{t}$ is the combined vector. To this end, we have obtained the aggregation vector $\mathbf{h_t}$ and the global preference vector from the embedding of special token [CLS], denoting as $\mathbf{x_c}$. Then we concatenate the two vectors and pass them to a feed-forward layer. Finally, dropout and L2 normalization tricks are employed after the FFN layer then we obtain temporal view embedding as:

$$\mathbf{T}(s) = \text{L2Norm}(\text{FFN}(\text{Concat}(\mathbf{h_t}, \mathbf{x_c}))). \tag{6}$$

4.2 Spatial Encoder for Session Graphs

The subsection shows the session graph construction process and its learning process, illustrated in the local spatial part of Fig. 3.

4.2.1 *Multi-relational Session Graph Construction*

There may exist duplicate items in one session. Thus, it is important to construct a session graph to capture such the spatial relationship in terms of item transitions. Given a session s with a *repeatable* item sequence $s = [v_1, v_2, v_3, \ldots, v_M]$, let $G_s = (V_s, E_s)$ be the corresponding session graph where the node set V_s consists the unique items in the session, the edge set E_s contains edges represented any two adjacent items (v_i, v_j) in the sequence s, forming an item-transition pattern behind the session.

Contrast to FGNN [12] which utilizes the occurrence frequency of edges to construct a weighted directed graph for a session, we leverage a multi-relational weighted graph which uses multiple types of relationship, including in-relation, out-relation, bi-direction and self-loop. Specifically, the out-relation indicates that there only exists a transition (v_i, v_j) in the graph, the in-relation is vice versa. The bi-direction represents that (v_i, v_j) simultaneously exits bi-directional transition. Besides, the self-loop implies that there exist a self transition of an item. By using these four relationships, the spatial structure can be enriched by more accurate inter-relationships among item transitions. We name this graph as Multi-relational Session Graph (MSG). A concrete example is demonstrated in Fig. 3, in which the session $s_1 = [v_1, v_2, v_3, v_2, v_4, v_5]$ can be converted into a multi-relational graph as shown inside the blue dotted rectangle with local.

4.2.2 *Multi-relational Graph Attention Network for MSGs*

We next present how to propagate item features on a multi-relational session graph to encode item-transitional relations. Graph attention network (GAT) [33] and Multi-relational GCN [34] have shown their powerful capability in graph structure and multiple types of edge relations learning, respectively. We further extend them to our multi-relational weighted graph and denote the model as MGAT.

The input to our encoder layer is a set of item features after embedding layer, $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \mathbf{h_3} \ldots, \mathbf{h_U}]$, where $\mathbf{h_i} \in \mathbb{R}^D$, U is the number of unrepeatable items in current session ($U \leq M$), and D is hidden size. We define relation embedding of in-relation, out-relation, bi-direction, and self-loop as $\mathbf{r_{in}}$, $\mathbf{r_{out}}$, $\mathbf{r_{bi}}$, and $\mathbf{r_{self}}$ respectively. We denote $\mathbf{r_{ij}}$ as a general relation embedding between v_i and v_j that is determined by the specific relation between the two items, *i.e.*, one of the four relations. The attention scores among these items are calculated by

$$e_{ij} = \mathbf{r_{ij}}^\mathsf{T} \left(\mathbf{h_i} \circ \mathbf{h_j}\right), \tag{7}$$

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}(e_{ij})\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}(e_{ik})\right)}, \tag{8}$$

where e_{ij} is the relational similarity between item v_i and its neighbor v_j by element-wise product and relational inner product, $\alpha_{i,j}$ is the the attention scores.

It is worth noting that our MGAT is different from [33], [34], [40], we employ a multi-head attention mechanism to incorporate all edge relations instead of a single head latent space to better enhance the representation ability for the spatial structure. To be specific, each head computes a kind of relations among items and their neighbors, and then the embeddings of multi-head attention are added rather than concatenated:

$$\tilde{\mathbf{h}}_\mathbf{i} = \sum_{r=1}^{R} \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(r)} \mathbf{h_j}, \tag{9}$$

where $R = 4$ denotes that four relations mentioned above, $\alpha_{i,j}^{(r)}$ are normalized attention coefficients of item v_i and its neighbor v_j in the r-th relation head. Then, we get the attention-aware representation $\tilde{\mathbf{H}} = \left[\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \tilde{\mathbf{h}}_3 \ldots, \tilde{\mathbf{h}}_M\right]$ of a specific session based on the initial item order of the session, where M is the item number of the current session.

4.2.3 *Local Spatial Aggregation*

To emphasize the recent preference within the current session, we concatenate $\tilde{\mathbf{H}}$ representations with a learnable position embedding $\mathbf{P}_s = [\mathbf{p_M}, \mathbf{p_{m-1}}, \mathbf{p_{m-2}} \ldots, \mathbf{p_1}]$. Besides, the session information can also be represented as the average in general. Thus, we take the two ways into consideration:

$$\check{\mathbf{H}} = \tanh(\text{Concat}(\mathbf{P_s}, \tilde{\mathbf{H}})\mathbf{W_s}), \tag{10}$$

$$\overline{\mathbf{H}}_\mathbf{s} = \frac{1}{M} \sum_{i=1}^{M} \tilde{\mathbf{H}}_\mathbf{i}, \tag{11}$$

$$\beta_\mathbf{s} = \text{sigmoid}\left(\check{\mathbf{H}}\mathbf{W_5} + \overline{\mathbf{H}}_\mathbf{s}\mathbf{W_6} + \mathbf{b_5}\right)\mathbf{f_s}, \tag{12}$$

where $\check{\mathbf{H}}$ is the position-sensitive session embedding, $\mathbf{H}_s$ is the general session embedding, $\beta_\mathbf{s}$ is soft-attention score indicating the importance of each item, and $\mathbf{W_s} \in \mathbb{R}^{2D \times D}, \mathbf{W_5}, \mathbf{W_6} \in \mathbb{R}^{D \times D}, \mathbf{b_5}, \mathbf{f_s} \in \mathbb{R}^D$ are trainable parameters. Finally, the spatial view embedding of a session s is calculated by combing item embeddings with their corresponding importance $\beta_\mathbf{s}$:

$$\mathbf{G}(s) = \sum_{i=1}^{M} \beta_{s i} \tilde{\mathbf{h}}_\mathbf{i}. \tag{13}$$

4.3 Contrastive Loss function

One of the key properties of contrastive learning is to align features from positive pairs [43]. Such positive pairs could be (i) a data sample with two augmentation tricks before being fed into a encoder [35], [41], (ii) a data sample with twice dropout noises in a encoder [42], or (iii) a data sample with two different encoders [52]. Inspired by the [52], which constructs the contrastive samples with two different encoders, we utilize contrastive learning that aims to align the augmented representations from the spatial encoder and the temporal encoder in the latent space to maximize the lower bound of mutual information of the two views.

To achieve the target, a spatio-temporal contrastive loss function is designed to distinguish whether the two representations are derived from the same session. Specifically, the contrastive loss learns to minimize the difference between the augmented spatial and temporal views of the same session and maximize the difference between the two augmented views derived from the different sessions. Technically, considering a mini-batch of C sessions s_1, s_2, $\ldots$, s_i, $\ldots$, $s_{|C|}$, we get the output embeddings from the spatial encoder (see Eq. 13) and the temporal encoder (see Eq. 6), denoted as $\mathbf{G}(s_i)$ and $\mathbf{T}(s_i)$ for each session, respectively, where we treat $(\mathbf{G}(s_i), \mathbf{T}(s_i))$ as the positive pair. For the negative samples, we propose a mixed noise negative sampling strategy which applies a column-wise shuffling operator for each $\mathbf{T}(s_i)$ in the batch to produce the noisy temporal samples and combine them with all $\mathbf{T}(s)$ to obtain a $2|C|$ negative candidate pool, then randomly samples $|C|$ negative examples denoted as C^- within the pool. Formally, inspired by SimCLR [35], we adopt InfoNCE [53] as contrastive loss that can be formulated as

$$\mathcal{L}_{cont} = -\sum_{i=1}^{|C|} \log \frac{\exp\left(\mathrm{sim}\left(\mathbf{G}(s_i), \mathbf{T}(s_i)\right)/\tau\right)}{\sum_{s^- \in C^-} \exp\left(\mathrm{sim}\left(\mathbf{G}(s_i), \mathbf{T}(s^-)\right)/\tau\right)}, \tag{14}$$

where $\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|}$ computes the cosine similarity, and τ is a fixed temperature parameter. By minimize the contrastive objective, we can obtain the enhanced session representations with sufficient interaction between spatial and temporal augmented views in the latent space.

5 MAIN SUPERVISED TASK OF RESTC

Note that the auxiliary contrastive learning task does not need labels. This section introduces the main supervised task to aggregate spatial and temporal embeddings. Since collaborative filtering information could also be in the format of graph, we construct the global collaborative filtering graph to enhance the spatial encoder (see details in Sec. 5.1). Sec. 5.2 illustrates how to generate the final session representation to fuse the temporal embeddings and the enhanced spatial embeddings, based on which RECTC predicts the next item (see Sec. 5.3). Lastly, Sec. 5.4 presents how to jointly train the contrastive and downstream tasks via a multi-task fashion.

5.1 Spatial Encoder for the CFG

A Collaborative Filtering Graph (CFG) is to learn the collaborative filtering information of a session based on a global item-transition view. Given a complete session set from all anonymous users, denoted as $S = [s_1, s_2, s_3, \ldots, s_l]$, let $G_{cf} = (V_{cf}, E_{cf})$ be a graph where $V_{cf} \in I$ denotes the item set and E_{cf} represents weighted edges from all item-relationships. We define that an item pair has a *connection* in a session if they are adjacent in such a session, the times of repeated *connections* are treated as the weight of the edge between the pair. This can be found in the global spatial encoding part of Fig. 3.

5.1.1 *Collaborative Filtering Graph Encoding*

Obtaining the embedding of CFG is to enrich the representation of a session with implicit collaborative filtering information from other session data. Without the assistance of CFG embeddings, modeling of a single short-term session could be ineffective in capturing complex transitional relationships among items overall sessions, and it will suffer from severe data sparsity problems. In such a case, we leverage the GraphSAGE-GCN [54], which used the mean-pooling propagation rule to subtly encode the CFG to aggregate K-hop neighbors' information of every item. The one layer of the encoder is:

$$\mathbf{Z}^{(k)} = \mathrm{LeakyReLU}(\tilde{\mathbf{D}}^{-\mathbf{1}}\tilde{\mathbf{A}}\mathbf{Z}^{(k-1)}\mathbf{W_c}^{(k)}), \tag{15}$$

where $\mathbf{Z}^{(0)} \in \mathbb{R}^{N \times D}$ represents initial input embedding of items of all sessions, $\mathbf{W_c}^{(k)} \in \mathbb{R}^{D \times D}$ denotes learnable weight matrix in the k-th layer, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I_N}$ means that adjacent matrix added with identity matrix, which can be seem as a self-loop of items in CFG. And $\tilde{\mathbf{D}}_{\mathbf{ii}} = \sum_j \tilde{\mathbf{A}}_{\mathbf{ij}}$ are degree matrix over CFG. After passing K layers graph convolution encoder, we get the K-hop CFG embedding represented as $\tilde{\mathbf{Z}} = \mathbf{Z}^{(K)} = \left[\mathbf{z_1}^{(K)}, \mathbf{z_2}^{(K)}, \mathbf{z_3}^{(K)} \ldots, \mathbf{z_N}^{(K)}\right]$, where N is the number of items overall sessions.

5.1.2 *Spatial Encoder Enhancing with CFG*

To obtain the enhanced graph-structure representation, we additionally add K-hop neighbor view from CFG (denoted as $\tilde{\mathbf{Z}}$), which is extracted from global CFG embeddings that involve items in the current session s (denoted as $\tilde{\mathbf{Z}}_{\mathbf{s}}$). The embedding of a specific session is:

$$\mathbf{H_g} = \mathrm{Concat}(\mathbf{P_e}, \tilde{\mathbf{H}}, \tilde{\mathbf{Z}}_{\mathbf{s}})\mathbf{W_g}, \tag{16}$$

where $\mathbf{W_g} \in \mathbb{R}^{3D \times D}$ is trainable parameter, $\mathbf{P_e}$ is the position embedding mentioned in Eq.10, $\tilde{\mathbf{H}}$ is the output embedding of MSG in Eq.9. To this end, we have obtained enhanced graph-based session embedding that simultaneously contain the spatial view of current session and global collaborative filtering from all sessions.

5.2 Embedding Fusion

After the session data pass through the spatial graph encoder and the temporal sequence encoder at the meantime, we obtain the distinct semantic representations from two views. Due to the limitations of each view, we apply the soft-attention mechanism to interact the enhanced spatial graph embeddings with temporal embeddings to acquire attentive vectors. The details are listed as follows:

$$\mathbf{H}'_{\mathbf{g}} = \tanh(\mathbf{H_g}\mathbf{W_f}), \tag{17}$$

$$\rho_{\mathbf{s}} = \mathrm{sigmoid}\left(\mathbf{H}'_{\mathbf{g}}\mathbf{W_7} + \mathbf{T}\mathbf{W_8} + \mathbf{b_7}\right)\mathbf{f_g}, \tag{18}$$

$$\mathbf{s_h} = \sum_{i=1}^{U} \rho_{s\,i}\left(\tilde{\mathbf{z}}_{\mathbf{s_i}} + \tilde{\mathbf{h}}_{\mathbf{i}}\right), \tag{19}$$

where $\mathbf{H}_g$ is the spatial embedding from Eq. (16), $\mathbf{T}$ is the temporal embedding from Eq. 6, $\tilde{\mathbf{z}}_{\mathbf{s_i}}$ indicates the CFG embedding of the v_i in session s, and $\tilde{\mathbf{h}}_{\mathbf{i}}$ denotes the MSG embedding of v_i, $\mathbf{W_f}, \mathbf{W_7}, \mathbf{W_8} \in \mathbb{R}^{D \times D}$ are learnable matrices, and $\mathbf{b_7}, \mathbf{f_g} \in \mathbb{R}^D$ are learnable biases. Finally, we get the semantic-rich representation $\mathbf{s_h}$ which considers the global collaborative filtering spatial, the session spatial, and the session temporal information.

TABLE 1: Dataset Statistics.

Dataset	Items	Clicks	Train	Test	Avg.len
Tmall	40,728	818,479	351,268	25,898	6.69
Diginetica	43,097	982,961	719,470	60,858	5.12
Gowalla	29,510	1,122,788	419,200	155,332	3.85
RetailRocket	36,968	710,586	433,648	15,132	5.43
Nowplaying	60,417	1,367,963	825,304	89,824	7.42
LastFM	38,615	3,835,706	2,837,330	672,833	11.78

5.3 Next-item Prediction Task

We further make use of the session embedding $\mathbf{S_h}$ to make recommendations by computing the probability distributions of the candidate items. Specifically, we utilize the softmax function to obtain the main task output:

$$\hat{\mathbf{y}} = \operatorname{softmax}\left(\mathbf{s_h}\mathbf{W}_y\right), \tag{20}$$

where $\mathbf{W}_y \in \mathbb{R}^{D \times N}$ is transformation matrix for the distribution prediction, $\hat{\mathbf{y}}_i$ represent the output probability of the prediction. Then, we apply cross-entropy as our objective function of the main task with the ground truth $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \ldots, \mathbf{y}_N\}$:

$$\mathcal{L}_{main} = -\sum_{i=1}^{N} \mathrm{y}_i \log\left(\hat{\mathbf{y}}_i\right) + \left(1 - \mathrm{y}_i\right) \log\left(1 - \hat{\mathbf{y}}_i\right) \tag{21}$$

5.4 Multi-task Training for Contrastive and Supervised Tasks

We unified the main recommendation task with the contrastive learning task to enhance the performance of SBR, which could be viewed as a multi-task training process:

$$\mathcal{L} = \mathcal{L}_{\text{main}} + \eta_1 \mathcal{L}_{cont} + \eta_2 \|\Theta\|_2^2, \tag{22}$$

where η_1 controls the strength of contrastive learning and η_2 is the constant of L_2 regularization of the all trainable parameters Θ.

6 Experiment

6.1 Experimental Settings

In this section, aiming to answer the following research question, we conduct extensive experiments on six datasets.

- **RQ1** How does RESTC perform, compared to present methods in the SBR task?
- **RQ2** Are the main components (e.g., Spatial encoder, Temporal encoder, CFG encoder, spatio-temporal contrastive learning) really working well?
- **RQ3** How does the spatial encoder (MGAT) work effectively compared to other GNN-based backbones?
- **RQ4** How do different settings (temperature τ, negative sampling strategies) of contrastive learning impact the performance of RESTC?
- **RQ5** Are RESTC robust to different lengths of session data?
- **RQ6** How do different hyper-parameters affect RESTC?
- **RQ7** Is the spatio-temporal contrastive learning really improving the representation learning?

6.1.1 Dataset Description

We evaluate our RESTC on six public benchmark datasets: *Tmall*[1], *Diginetica*[2], *Gowalla*[3], *RetailRocket*[4], *Nowplaying*[5], *LastFM*[6], which are often used in session-based recommendation models. **Tmall** comes from a competition in IJCAI, which contains anonymous users' shopping logs on the Tmall online website. **Diginetica** records the clicks of anonymous users within six months, and it is from the CIKM Cup platform 2016. **Gowalla** is a check-in dataset that is widely utilized by point-of-interest recommendation. We follow [14] to process this data. **RetailRocket** is original from a Kaggle contest published by an e-commerce company, which contains the browser activity of anonymous users within six months. **Nowplaying** describes the music listening behavior of users, and it comes from the resource of [55]. **LastFM:** is a popular music dataset that has been used as a benchmark in many recommendation tasks. Following [56], we employ it as session-based data.

Moreover, we adopt the data augmentation and filtering for the sessions following by [5], [6], [12], [13]. Specifically, we process these datasets into sessions. Concretely, we get rid of all sessions whose length is shorter than 1 and the appearing of items less than 5 overall sessions. We also set the data of last 7 days to be the test data and the previous data as train data. In addition, given a session data $s = [v_1, v_2, \ldots, v_M]$, we augment the sequence and generate corresponding labels by splitting it into $([v_1], v_2), ([v_1, v_2], v_3), \ldots, ([v_1, v_2, \ldots, v_{m-1}], v_M)$ for all sessions in six datasets. The details of processed data are shown in Table 1.

6.1.2 Baselines

- **FPMC** [36] learns the representation of session via Markov-chain based method. We ignore the user profile information in the experiment and adapt it to the session-based recommendation.
- **GRU4Rec** [10] is an RNN-based method that utilizes GRU and adopts ranking-based loss to the model preference of users within the current session.
- **NARM** [4] is a attention-based RNN model to learn session embedding.
- **STAMP** [11] is an attention model to capture user's temporal interests from historical clicks in a session and relies on self-attention of the last item to represent users' short-term interests.
- **SR-GNN** [5] is the first GNN-based model for the SBR task, which transforms the session data into a direct unweighted graph and utilizes gated GNN to learn the representation of the item-transitions graph.
- **GC-SAN** [6] uses gated GNN to extract local context information and then employs the self-attention mechanism to obtain the global representation.

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42
[2] http://2015.recsyschallenge.com/challenge.html
[3] https://snap.stanford.edu/data/loc-gowalla.html
[4] https://www.kaggle.com/retailrocket/ecommerce-dataset
[5] http://dbis-nowplaying.uibk.ac.at/
[6] http://ocelma.net/MusicRecommendationDataset/lastfm-1K.html

TABLE 2: The comparison over all datasets. ⋆ indicates a statistically significant level p-value <0.001 comparing our RESTC with the baselines. Underlined numbers mean best baseline. The best performance for each benchmark is marked in black bold. TM, DG, RR, LF, NP, GW denote Tmall, Dignetica, RetailRocket, LastFM, Nowplaying and Gowalla, respectively.

Dataset	Metric	FPMC	GRU4REC	NARM	STAMP	SR-GNN	CSRM	FGNN	GC-SAN	GCE-GNN	TASRec	S^2-DHCN	**RESTC**	Imprv.
TM	HR@10	13.10	9.47	19.17	22.63	23.41	25.54	20.67	24.78	28.01	25.72	26.22	**35.57**	26.99%
	HR@20	16.06	10.93	23.30	26.47	27.57	29.46	25.24	28.72	33.42	29.58	31.42	**42.47**	27.08%
	MRR@10	7.12	5.78	10.42	13.12	13.45	13.62	10.67	13.55	15.08	14.22	14.60	**18.05**	19.69%
	MRR@20	7.32	5.89	10.70	13.36	13.72	13.96	10.39	13.43	15.42	14.51	15.05	**18.52**	20.10%
DG	HR@10	15.43	17.93	35.44	33.98	36.86	36.59	37.72	37.86	41.16	39.85	40.21	**42.35**	2.89%
	HR@20	26.53	29.45	49.70	45.64	50.73	50.55	50.58	50.84	54.22	52.53	53.66	**55.93**	3.15%
	MRR@10	6.20	7.33	15.13	14.26	15.52	15.41	15.95	16.89	18.15	17.19	17.59	**18.75**	3.31%
	MRR@20	6.95	8.33	16.17	14.32	17.59	16.38	16.84	17.79	19.04	18.22	18.51	**19.65**	3.20%
RR	HR@10	25.99	38.35	42.07	42.95	43.21	43.47	43.75	43.53	48.22	46.32	46.15	**50.12**	3.94%
	HR@20	32.37	44.01	50.22	50.96	50.32	51.02	50.99	50.71	55.78	54.23	53.66	**57.81**	3.64%
	MRR@10	13.38	23.27	24.88	26.41	26.07	25.58	26.11	26.03	28.36	27.22	26.85	**30.15**	6.31%
	MRR@20	13.82	23.67	24.29	25.17	26.57	26.19	26.21	25.76	28.72	28.37	27.30	**30.82**	7.31%
LF	HR@10	6.65	11.21	15.37	14.99	15.12	15.47	15.32	15.68	17.22	16.83	17.09	**18.57**	7.84%
	HR@20	12.91	17.79	21.86	22.06	22.29	22.31	22.18	22.64	24.05	23.22	22.86	**25.54**	6.20%
	MRR@10	3.21	4.79	7.12	7.27	7.19	7.33	7.09	7.62	7.74	8.22	8.02	**8.87**	7.91%
	MRR@20	3.73	5.41	7.55	7.84	8.31	8.12	8.03	8.42	8.19	8.65	8.45	**9.28**	7.28%
NP	HR@10	5.28	6.74	13.60	13.22	14.17	13.20	13.89	14.11	16.94	16.35	17.35	**18.39**	5.99%
	HR@20	7.36	7.92	18.59	17.66	18.87	18.14	18.75	19.19	22.37	20.52	23.50	**24.79**	5.49%
	MRR@10	2.68	4.40	6.62	6.57	7.15	6.08	6.80	7.11	8.03	7.37	7.87	**8.31**	3.49%
	MRR@20	2.82	4.48	6.93	6.88	7.74	6.42	7.15	7.54	8.40	7.78	8.18	**8.72**	3.91%
GW	HR@10	20.47	31.56	40.53	40.99	41.89	42.11	42.09	42.17	44.25	43.21	45.11	**47.86**	6.10%
	HR@20	29.91	41.91	50.11	50.15	50.29	50.17	50.11	50.71	52.48	53.55	53.34	**56.38**	5.70%
	MRR@10	9.88	17.85	22.94	23.10	23.78	23.33	22.91	23.77	24.11	23.19	23.29	**25.33**	5.06%
	MRR@20	11.37	18.29	23.89	24.03	24.31	24.23	24.11	24.58	24.68	23.73	23.88	**25.92**	5.02%

- **CSRM** [57] integrates an internal memory encoder through an external memory network by considering the correlation between neighboring sessions.
- **FGNN** [12] proposes to leverage a weighted graph attention network for computing the information flow in the session graph and generates the user preference by a graph readout function.
- **GCE-GNN** [40] transforms the sessions into global graph and local graphs to enable cross session learning.
- **TASRec** [18] incorporate temporal information via constructing a sequence of dynamic graph snapshots at different timestamps.
- **S^2-DHCN** [49] transforms the session data into hypergraph and line-graph and and uses self-supervised learning to enhance session-based recommendation.

6.1.3 Evaluation Merics and Parameter Settings

Following the baselines mentioned above, we adopt two widely used metrics for SBR task: **HR@N** (Hit Rate) and **MRR@N** (Mean Reciprocal Rank). We report their optimal performance for each baseline following the original setting from their papers. In our settings, we apply grid search to find the optimal parameters based on the random 20% of train data as validation. Concretely, we search the embedding dimension from the range $\{100, 150, 200, 250, 300, 350\}$, and the default batch size is set to 512. We also investigate the coefficient of the contrastive learning task from 5e-4 to 1e-1. The default constant of L_2 regularization is 1e-5 in our experiments. We stack 2 SESTrans encoder layers as default, which achieve best performance to capture temporal pattern in our experiments. Then we search the MGAT and CFG encoding layers from 1 to 4, we find that 1 MGAT layer and 3 CFG embedding layers is already enough for learning the spatial structure representation of a session and the experimental details are shown on Sec 6.7 Besides, we utilize the Adam optimizer with the learning rate of 0.001 as well as Step-LR and Cosine-Annealing-LR schedulers to adjust learning rate.

6.2 Overall Results (RQ1)

The experiment results of baselines and RESTC model over six datasets are reported in Table 2. The performance results show that traditional machine learning method FPMC is worse than deep learning methods since it cannot capture long-time dependency. For sequence-based methods, STAMP and NARM perform better than GRU4REC since they utilize attention mechanisms to learn the critical relations among all items. Besides, CSRM performs the best among sequence-based baselines, demonstrating the effectiveness of leveraging collaborative filtering information from other sessions. Besides CSRM performs the best compared with STAMP and NARM, demonstrating the effectiveness of leveraging collaborative filtering information from other sessions.

Note that GNN-based methods outperform sequence-based methods, which indicates that there still exists some functional yet undiscovered spatial-structure patterns in sequence-based methods; Moreover, information on item-transition graphs (in the spatial view) might be relatively more informative than temporal view as in sequence-based methods. Specifically, GC-SAN shows better results than SR-GNN, demonstrating that combining GNN with self-attention could better model the current session's local and global context information. GCE-GNN shows better results than SR-GNN and GC-SAN, demonstrating that combining the information of the local session and global neighbor graphs is effective to enrich the session representation. TASRec outperforms general GNN-based methods like SR-GNN, FGNN and GC-SAN, proving that incorporating temporal information is significant to spatial structure. S^2-DHCN shows great performance in LastFM and Gowalla in term of HR@20 since it makes use of both inter-

and intra-relations overall sessions and then applies self-discrimination to improve the representation.

As for our RESTC, the results show that our proposed method significantly outperforms all comparative baselines, including sequence-based, GNN-based, temporal-enhanced and contrastive learning based methods. Especially, compared with all the baselines, RESTC has an obvious improvement on Tmall as 27.08% on HR@20 and 20.10% on MRR@20, which reflects RESTC's superior representation capability. In particular, the significant improvement of RESTC over strong baselines (e.g., GCE-GNN and S^2-DHCN) implies that leveraging temporal and collaborative filter information is potential for refining the session representation. Besides, RESTC outperforms temporal-enhanced GNN like TASRec with a large margin, indicating that adequate interaction between spatial and temporal views via contrastive learning can significantly boost the performance.

TABLE 3: Ablation Study in Variants of RESTC.

Dataset	**TM**		**DG**		**RR**	
Measures	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR @20
w/o SESTrans	36.61	16.08	54.31	19.11	52.65	27.11
w/o CFG	36.96	16.38	54.28	18.84	56.28	28.94
w/o Cont.	39.27	17.11	54.65	19.35	57.01	29.13
w/o PE-G	40.81	17.75	51.98	18.04	53.51	27.96
w/o PE-S	41.05	17.92	54.45	19.29	56.98	30.03
RESTC	**42.47**	**18.52**	**55.93**	**19.65**	**57.81**	**30.82**

6.3 Ablation Study (RQ2)

We further investigate the effectiveness of each module in our RESTC model by conducting Ablation experiments. Concretely, we design several contrast variants of RESTC, they are: (i) w/o SESTrans, which removes the temporal encoder SESTrans thus without the spatio-temporal contrastive learning; (ii) w/o CFG, which only considers the spatial encoder MGAT and spatial encoder SESTrans, without the CFG embedding; (iii) w/o Cont, which contains two complete augmented encoders without the contrastive learning task. Besides, to investigate the impact of position embedding for the spatial and temporal views, (iv) w/o PE-G and w/o PE-S represent RESTC model without learnable position embedding in spatial encoder MGAT and without timeline absolute position embedding in temporal encoder SESTrans.

From Table 3, we can clearly observe that removing the above components consistently leads to performance drop, implying that these components are all significant to RESTC. Concretely, w/o SESTrans underperforms w/o Cont, showing that incorporating temporal information through directly combining the temporal embedding with spatial embedding in the main supervised task has already improve the performance. Then, the downward trend of w/o CFG is more obvious than w/o Cont. This is consistent with our assumption that obtaining the implicit collaborative filtering information from the global weighted session graph, denoted as CFG, is able to enhance spatial representation, which can help to remedy data sparsity problem for the short-term session. Furthermore, it can be observed that spatio-temporal contrastive learning enhances the performance on both metrics by comparing standard RESTC with RESTC w/o Cont with obvious margin. This reveals that cross-view interaction via contrastive regularization in the latent space can further reinforce the session representation for the main prediction task.

Besides, w/o PE-G demonstrates that removing the position embedding in the spatial structure view results in a remarkable performance drop since the model cannot recover the initial order relation after graph embedding. Moreover, w/o PE-S performs worse than RESTC in the selective datasets and shows that the effectiveness of temporal-aware encoding in the temporal encoder SESTrans.

6.4 Comparison with Different Spatial Encoder backbones (RQ3)

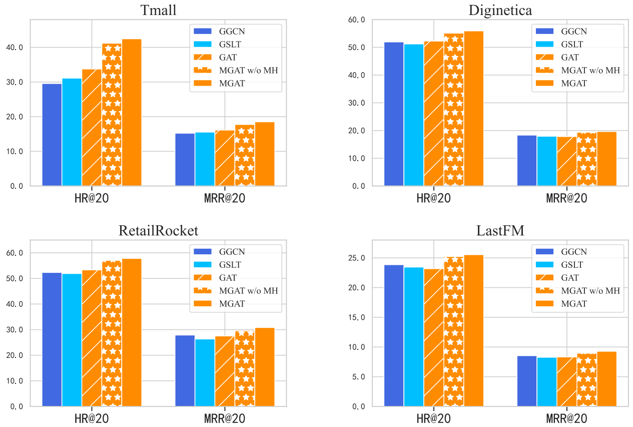
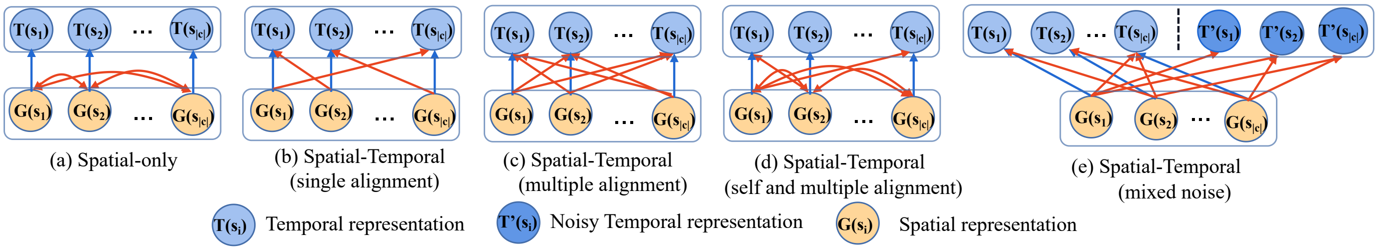


Fig. 4: Results of RESTC with different GNN-based spatial encoders. GSLT denotes GraphSAGE-LSTM, MGAT w/o MH denotes the single-head MGAT.

Since our proposed RESTC is an model-agnostic framework that can effectively adapt to various of GNN-based spatial encoders, we want to further investigate the effectiveness of leveraging MGAT to learn the spatial representation of the session graph. Therefore, we compare it with other GNN-based backbones on Tmall, Diginectica, RetailRocket and LastFM. Specifically, we substitute MGAT backbone with some variants, including Graph Gate Neural Network (GGNN) [5], [6], GraphSAGE-LSTM [54], GAT [12], [33] and MGAT without multi-heads attention. Among them, GGCN constructs the session as a weighted directed graph and uses the occurrence frequency of item-pair transitions as edges and applies gate-based aggregate function; GraphSAGE-LSTM and GAT also adopt the same method to construct the session graph, but they utilize LSTM and attention weighted sum as the aggregation functions, respectively. As depicted in the Figure 4, RESTC equipped with MGAT as spatial encoder is obviously superior than all the comparative GNN-based backbones. Concretely, MGAT backbone has a significant improvement compared with GAT and MGAT w/o MH, verifying the advantage of constructing sessions as multi-relational session graphs and leveraging multi-head MGAT. Moreover, compared to the GraphSAGE-LSTM and GGCN, MGAT achieves better performance which suggests that the attention-mechanism is more powerful for learning the spatial structural representation for the session graph.


T(s1)
T(s2)
…
T(s|c|)
G(s1)
G(s2)
…
G(s|c|)
T'(s1)
T'(s2)
T'(s|c|)
T(si) Temporal representation
T'(si) Noisy Temporal representation
G(si) Spatial representation

(a) Spatial-only

(b) Spatial-Temporal (single alignment)

(c) Spatial-Temporal (multiple alignment)

(d) Spatial-Temporal (self and multiple alignment)

(e) Spatial-Temporal (mixed noise)

Fig. 5: Four variants of negative sampling strategy and the default method.

6.5 Further Analysis on Spatio-Temporal Contrastive Learning (RQ4)

To further analyze what factors affect the performance of our proposed spatio-temporal contrastive learning, we move on to studying different settings. We first investigate the impact of temperature τ. Then, we dive into the influence of distinct negative sampling strategies in the contrastive learning objective function. We adjust the hyperparameter τ on Tmall and Diginetica which have a similar trend to other datasets and demonstrate the results of using variants of negative sampling on Tmall, Diginetica, RetailRocket and LastFM due to the limited space.

6.5.1 Impact of Temperature τ

As mentioned in [32], [35], τ play a critical role in hard negative mining for contrasative learning. The experiment results in Fig 6 show the curves of RESTC performance with respect to different τ. We can observe that: (1) The larger the value of τ (e.g., 1.0), the slower the model converges during training, and there is a significant decrease for the performance of the model when it converges. Similar to [32], we attribute this phenomenon to the difficulty of identifying hard negative samples, whose temporal representations are similar to that of positive samples, thus making the model fail to distinguish them from the positive samples in the latent space. (2) In contrast, adjusting τ with a too small value (e.g., 0.1) will cause the model to converge quickly and lead to overfitting prematurely during training, since the small τ could make the model focus excessively on the hard negative samples and offer more gradients to guide the optimization, thus making the spatial and temporal representations easier to discriminate then accelerate the training process [58]. Therefore, we choose the value of τ between 0.1 and 1 depending on the dataset.

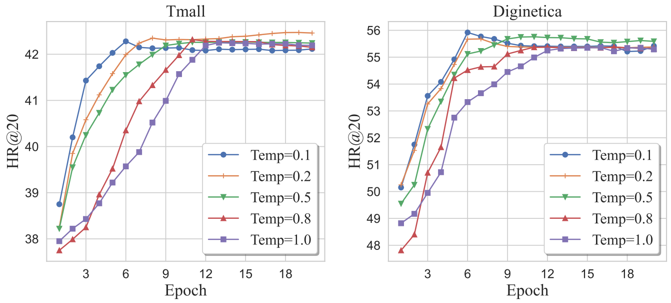


Fig. 6: Model performance of RESTC with different temperature τ.

6.5.2 Variants of Negative Sampling Strategy

To investigate how the choices of negative sampling affect the performance of contrastive learning, we ablate on several negative sampling strategies as show in Fig 5. Specifically, we compare our default method with four variants of session-level contrastive learning, which select negative samples from spatial or temporal session representations in a training batch: (a) *Spatial-only*, which selects the representations of other sessions in the spatial candidates *Spatial-only*; (b) *Spatio-Temporal (single alignment)*, which randomly selects one different temporal presentation, denoted as S-T (sa) ; (c) *Spatio-Temporal (multiple alignments)*, which selects the other temporal representations from the batch, denoted as S-T (ma); (d) *Spatio-Temporal (self and multiple alignments)*, which selects the representations of both saptial and temporal candidates in the batch, denoted as S-T (sma). As illustrated in Sec 4.3, the default method of RESTC is *Spatio-Temporal (mixed noise)*.

From Table 4 we can observe that, Spatial-only method performs worse than all the comparative methods, which only uses spatial representations as negative samples. This indicates that without using temporal representations as negative samples, it will be difficult to align spatio-temporal information in the latent space then lead to sub-optimal performance. Besides, S-T (ma) and S-T (sma) slightly perform better than S-T (sa), which we conjecture is because, increasing the sampling size and diversity of negative samples (spatial and temporal views) facilitates the model to distinguish between positive and negative sample pairs. In addition, S-T (mn) outperforms all the variants of sampling strategies, which may be because adding random noise to the set of temporal representation is beneficial to enhance the robustness of contrastive learning. Moreover, we also validate the correlation of noise sampling strategy and bach size. The results are consistent with SimCLR [14], increasing the number of negative samples by increasing the batch size frpm 128 to 512 significantly improves performance.

TABLE 4: Comparison on Variants of Negative Sampling

Dataset	**TM**		**DG**		**RR**	
Measures	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR @20
Spatial-only	41.33	17.96	54.95	19.39	57.19	30.15
S-T (sa)	41.95	18.11	55.33	19.44	57.35	30.19
S-T (ma)	42.23	18.18	55.56	19.42	57.45	30.52
S-T (sma)	42.35	18.23	55.52	19.45	57.38	30.44
S-T (mn)(bz=128)	41.95	18.15	55.32	19.44	57.23	30.32
S-T (mn)(bz=256)	42.22	18.21	55.56	19.53	57.41	30.56
S-T (mn)(bz=512)	**42.47**	**18.52**	**55.93**	**19.65**	**57.81**	**30.82**

6.6 Analysis on Different Session Lengths (RQ5)

In many scenarios, sessions are transferred to the server at various lengths [59]. It is worthwhile to investigate the

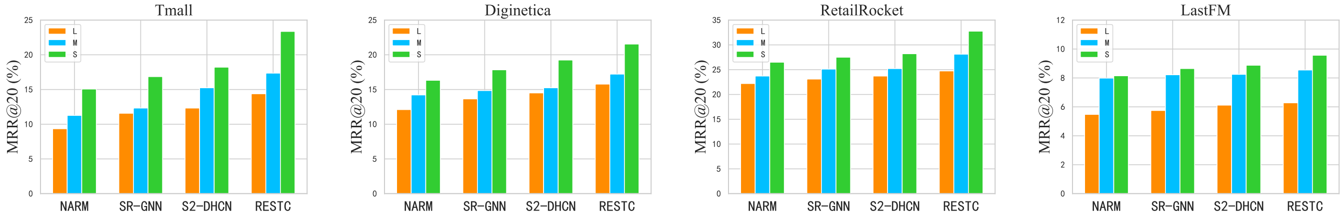
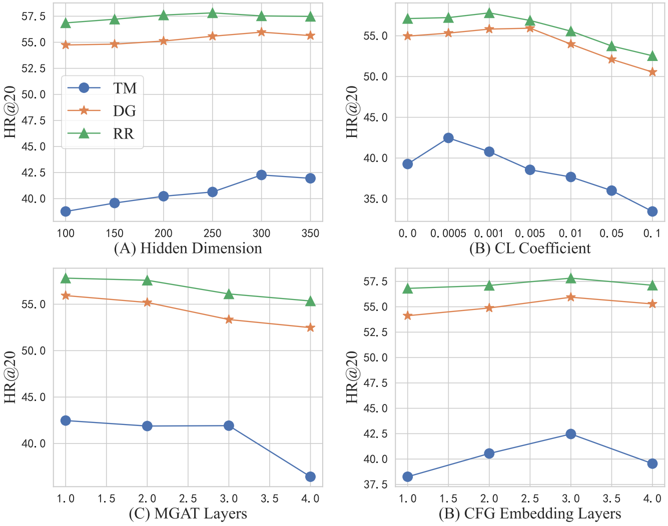


Fig. 7: MRR@20 on sessions of different lengths.

robustness of our RESTC model compared with baselines on different length of sessions. We separate all the sessions in Tmall, Diginetica, RetailRocket and LastFM into three groups, **short group** (S) with length of sessions from 0 to 5, **medium group** (M) with sessions from 5 to 10, rest of sessions are in the **long group** (L). We utilize MRR@20 to evaluate the performance of the methods instead of HR@20, since the MRR metric can better reflect the ranking quality of correct results. Fig. 7 demonstrates that RESTC outperforms the selective squence-based baseline NARM, the GNN-based baseline SR-GNN, the contrastive learning augmented GNN baseline S^2-DHCN with different lengths of sessions. Note that all methods have performance drops when session length increases. This may be because that long item-transitions are difficult to model users' preferences since the diversity of user's intents or missed clicks in the long sequence. Besides, the reuslts indicates that the superiority of RESTC in the scenarios when the ongoing session is short because of its effectiveness of handling data sparsity thanks to the CFG embedding in Sec. 5.1.



(A) Hidden Dimension

(B) CL Coefficient

(C) MGAT Layers

(B) CFG Embedding Layers

Fig. 8: Hyperparameter analysis of RESTC.

6.7 Impact of Hyperparameters (RQ6)

Next, we analyze the sensitivity of RESTC with different hyperparameter settings. We only show the result of HR@20 on Tmall, Diginetica, Retailrocket, due to the limited space.

6.7.1 *Impact of Hidden Dimension*

To investigate the impact of hidden dimension, we test the performance when increasing the dimension from 100 to 400. From the leftmost of Fig. 8 (A), we can draw a conclusion that increasing the hidden dimension does not continuously increase the performance. Our RESTC model achieves the best performance in 300 for Diginetica and Tmall, while obtaining an optimal result in 200 for Retailrocket. The reason might be that a larger hidden size might lead to overfitting.

6.7.2 *Strength of Contrastive Learning*

In RESTC, we utilize the hyperparameter η_1 to trade off the contrastive loss and the cross entropy loss. To demonstrate the utility of η_1, we compare the experimental results by using the η_1 values from $[0.0, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1]$. As in the rightmost of Fig 8 (B), larger η_1 does not show a tendency of better performance. Our model obtains the most satisfactory performance when η_1 is near 0.005, 0.001, 0.0005 for Diginetia, Retailrocket and Tmall, respectively. The HR@20 drops obviously when the η_1 becomes larger than these values, especially in Tmall. The main reason is that increasing η_1 might harm the optimization of the main prediction task. Therefore, we set the corresponding coefficient η_1 according to grid search.

6.7.3 *Effect of MGAT Layer*

To further analyze the impact of the aggregation layer numbers of the spatial encoder MGAT, we vary the number of MGAT layers in the range of $\{1, 2, 3, 4\}$. As the results presents in Fig 8 (C), leveraging 1 layer MGAT for RESTC has already achieve best performance and stacking more layers leads to decreasing tendency. We conjecture that adopting more layers will cause over-fitting issue since most of sessions are relatively short according to average lengths of the dataset statistics in Table 1.

6.7.4 *Effect of CFG Embedding Layer*

The embedding of a Collaborative Filtering Graph (CFG) enriches the current session with inter-session information, which is an efficient way to solve data sparsity problems and enhance the performance of recommendation. We range the layer numbers from 1 to 4 to study the impacts of the CFG embedding module's depth. From the middle of Fig. 8 we observe that the three-layer setting makes RESTC obtain the best result. And stacking more layers will add more noise information since over-smoothing issue of high-order relations of graphs.

6.8 Representation Quality of RESTC (RQ7)

To evaluate whether the spatio-temporal contrastive learning affects the representation learning performance, we utilize t-SNE to reduce the dimension of learned embeddings and visualize them in 2D planes. As shown in Fig. 9, we compare the visualize results of RESTC, RESTC(w/o

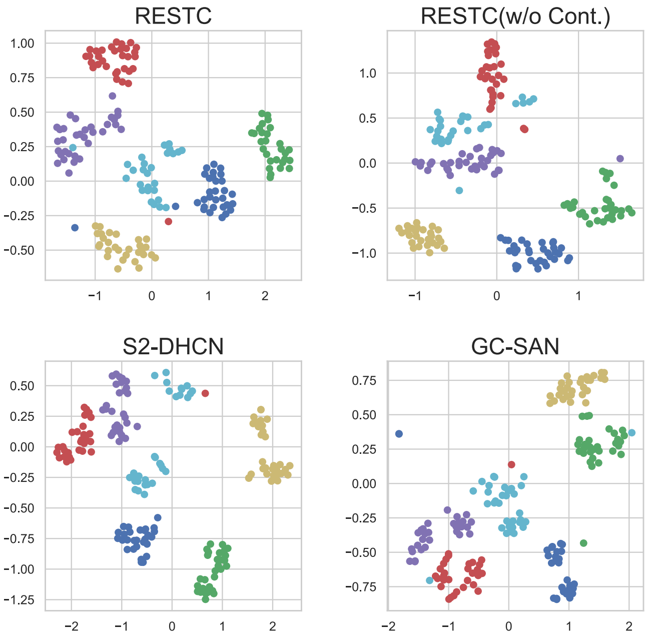


Fig. 9: t-SNE visualization of session embedding in a latent space, each color represents a specific label.

Cont.), S^2-DHCN and GC-SAN on Retailrocket and leverage six labels and randomly sample 50 session instances for each label. It is expected that session embeddings should be closer if they have the same label (next-to-click item). From Fig. 9, by comparing RESTC and its variant, we observe that removing spatio-temporal contrastive learning makes the learned embedding more indistinguishable in the latent space, showing that contrastive learning make a better alignment for RESTC between session embeddings w.r.t. the same label. Moreover, some session embeddings with different same labels are mixed to some degree for S^2-DHCN and GC-SAN, which make them indiscernible. In contrast, our RESTC shows a more diverse distribution and hence can better make correct prediction, demonstrating that superiority of RESTC in better representation learning.

7 Conclusion

This paper proposes a novel framework called RESTC, which aims to effectively learn the session representation from cross-view interaction and collaborative filtering information. It is equipped with the spatio-temporal contrastive learning to extract self-supervised signals from spatial and temporal views to mitigate temporal information loss and improve the quality of represenation learning. In the next-item prediction task, we utilized the embedding of the collaborative filtering graph to enrich the spatial structure information, which can also solve the data sparsity problem of the short-term session. Extensive experiment results demonstrate that RESTC achieves significant improvements compared with other recent baselines.

References

[1] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in *WWW*, 2017.

[2] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, 2019.

[3] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Comput. Surv.*, 2022.

[4] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *CIKM*, 2017, pp. 1419–1428.

[5] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *AAAI*, 2019, pp. 346–353.

[6] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *IJCAI*, 2019, pp. 3940–3946.

[7] Y. Chen, L. Huang, C. Wang, and J. Lai, "Hybrid-order gated graph neural network for session-based recommendation," *IEEE Trans. Ind. Informatics*, pp. 1458–1467, 2022.

[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *DLRS@RecSys*, 2016.

[9] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *CIKM*, 2018, pp. 843–852.

[10] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[11] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: short-term attention/memory priority model for session-based recommendation," in *KDD*, 2018, pp. 1831–1839.

[12] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *CIKM*, 2019, pp. 579–588.

[13] A. Luo, P. Zhao, Y. Liu, F. Zhuang, D. Wang, J. Xu, J. Fang, and V. S. Sheng, "Collaborative self-attention network for session-based recommendation," in *IJCAI*, 2020, pp. 2591–2597.

[14] T. Chen and R. C. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *KDD*, 2020, pp. 1172–1180.

[15] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *SIGKDD*, 2019.

[16] Z. Fan, Z. Liu, J. Zhang, Y. Xiong, L. Zheng, and P. S. Yu, "Continuous-time sequential recommendation with temporal graph collaborative transformer," in *CIKM*, 2021.

[17] Z. Pan, W. Chen, and H. Chen, "Dynamic graph learning for session-based recommendation," *Mathematics*, 2021.

[18] H. Zhou, Q. Tan, X. Huang, K. Zhou, and X. Wang, "Temporal augmented graph neural networks for session-based recommendations," in *SIGIR*. ACM, 2021.

[19] Q. Shen, S. Zhu, Y. Pang, Y. Zhang, and Z. Wei, "Temporal aware multi-interest graph neural network for session-based recommendation," *ArXiv*, vol. abs/2112.15328, 2021.

[20] J. Li, R. R. Selvaraju, A. Gotmare, S. R. Joty, C. Xiong, and S. C. Hoi, "Align before fuse: Vision and language representation learning with momentum distillation," in *NIPS*, 2021. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/505259756244493872b7709a8a01b536-Abstract.html

[21] Y. Zeng, X. Zhang, and H. Li, "Multi-grained vision language pre-training: Aligning texts with visual concepts," in *ICML*, 2021. [Online]. Available: https://arxiv.org/abs/2111.08276

[22] M. Liu and O. Tuzel, "Coupled generative adversarial networks," in *NIPS*, 2016, pp. 469–477.

[23] A. Blum and T. M. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT*, 1998, pp. 92–100.

[24] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. L. Yuille, "Deep co-training for semi-supervised image recognition," in *ECCV*, 2018.

[25] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *WSDM*, K. S. Candan, H. Liu, L. Akoglu, X. L. Dong, and J. Tang, Eds., 2022.

[26] N. Saunshi, O. Plevrakis, S. Arora, M. Khodak, and H. Khandeparkar, "A theoretical analysis of contrastive unsupervised representation learning," in *ICML*, 2019.

[27] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," 2021.

[28] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. J. McAuley, and C. Xiong, "Contrastive self-supervised sequential recommendation with robust augmentation," *CoRR*, vol. abs/2108.06479, 2021.

[29] Y. Chen, Z. Liu, J. Li, J. J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," in *WWW*, 2022.

[30] C. Wang, Y. Liang, Z. Liu, T. Zhang, and P. S. Yu, "Pre-training graph neural network for cross domain recommendation," *CoRR*, vol. abs/2111.08268, 2021.

[31] C. Huang, J. Chen, L. Xia, Y. Xu, P. Dai, Y. Chen, L. Bo, J. Zhao, and J. X. Huang, "Graph-enhanced multi-task learning of multi-level transition dynamics for session-based recommendation," in *AAAI*, 2021.

[32] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*, 2021, pp. 726–735.

[33] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[34] Z. Huang, X. Li, Y. Ye, and M. K. Ng, "MR-GCN: multi-relational graph convolutional networks based on generalized tensor product," in *IJCAI*, C. Bessiere, Ed., 2020.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*, vol. 119, 2020, pp. 1597–1607.

[36] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[37] W. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *IEEE International Conference on Data Mining*, 2018, pp. 197–206.

[38] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *CIKM*, 2019, pp. 1441–1450.

[39] J. Yuan, Z. Song, M. Sun, X. Wang, and W. X. Zhao, "Dual sparse attention network for session-based recommendation," in *AAAI*, 2021, pp. 4635–4643.

[40] Z. Wang, W. Wei, G. Cong, X. Li, X. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *SIGIR*, 2020, pp. 169–178.

[41] X. Chen, S. Xie, and K. He, "An empirical study of training self-supervised vision transformers," 2021.

[42] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[43] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International Conference on Machine Learning*. PMLR, 2020, pp. 9929–9939.

[44] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: graph contrastive coding for graph neural network pre-training," in *KDD*, 2020, pp. 1150–1160.

[45] T. Yao, X. Yi, D. Z. Cheng, F. X. Yu, T. Chen, A. K. Menon, L. Hong, E. H. Chi, S. Tjoa, J. J. Kang, and E. Ettinger, "Self-supervised learning for large-scale item recommendations," in *CIKM*, 2021, pp. 4321–4330.

[46] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *CIKM*, 2020, pp. 1893–1902.

[47] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T. Chua, "Contrastive learning for cold-start recommendation," in *ACM MM*, 2021, pp. 5382–5390.

[48] H. Li, X. Luo, Q. Yu, and H. Wang, "Session-based recommendation via contrastive learning on heterogeneous graph," in *IEEE Big Data*, 2021, pp. 1077–1082.

[49] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *AAAI*, 2021, pp. 4503–4511.

[50] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *NAACL-HLT*, 2019, pp. 4171–4186.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[52] K. Hassani and A. H. K. Ahmadi, "Contrastive multi-view representation learning on graphs," in *ICML*, 2020, pp. 4116–4126.

[53] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, ser. JMLR Proceedings, vol. 9, 2010, pp. 297–304.

[54] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *NIPS*, 2017.

[55] E. Zangerle, M. Pichl, W. Gassler, and G. Specht, "#nowplaying music dataset: Extracting listening behavior from twitter," in *WISMM*, 2014, pp. 21–26.

[56] L. Guo, H. Yin, Q. Wang, T. Chen, A. Zhou, and N. Q. V. Hung, "Streaming session-based recommendation," in *KDD*, 2019, pp. 1569–1577.

[57] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke, "A collaborative session-based recommendation approach with parallel memory modules," in *SIGIR*, 2019, pp. 345–354.

[58] S. Rendle and C. Freudenthaler, "Improving pairwise learning for item recommendation from implicit feedback," in *WSDM*, 2014.

[59] Z. Pan, F. Cai, W. Chen, and H. Chen, "Graph co-attentive session-based recommendation," *ACM Trans. Inf. Syst.*, 2022.