mathematics

MDPI

Article

Spatio-Temporal Contrastive Heterogeneous Graph Attention Networks for Session-Based Recommendation

Fan Yang * and Dunlu Peng

School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; pengdl@usst.edu.cn

* Correspondence: fany1993usst@gmail.com

Abstract: The main goal of session-based recommendation (SBR) is to analyze the list of possible next interaction items through the user's historical interaction sequence. The existing session recommendation models directly model the session sequence as a graph, and only consider the aggregation of neighbor items based on spatial structure information, ignoring the time information of items. The sparsity of interaction sequences also affects the accuracy of recommendation. This paper proposes a spatio-temporal contrastive heterogeneous graph attention network model (STC-HGAT). The session sequence is built as a spatial heterogeneous hypergraph, a latent Dirichlet allocation (LDA) algorithm is used to construct the category nodes of the items to enhance the contextual semantic information of the hypergraph, and the hypergraph attention network is employed to capture the spatial structure information of the session. The temporal heterogeneous graph is constructed to aggregate the temporal information of the item. Then, the spatial and temporal information are fused by sumpooling. Meanwhile, a modulation factor is added to the cross-entropy loss function to construct the adaptive weight (AW) loss function. Contrastive learning (CL) is used as an auxiliary task to further enhance the modeling, so as to alleviate the sparsity of data. A large number of experiments on real public datasets show that the STC-HGAT model proposed in this paper is superior to the baseline models in metrics such as *P*@20 and *MRR*@20, improving the recommendation performance to a certain extent.

Keywords: session-based recommendation; latent Dirichlet allocation; spatio-temporal heterogeneous; contrastive learning

MSC: 68T07

Citation: Yang, F.; Peng, D. Spatio-Temporal Contrastive Heterogeneous Graph Attention Networks for Session-Based Recommendation. *Mathematics* **2024**, *12*, 1193. https://doi.org/10.3390/math12081193

Academic Editor: Pasquale De Meo

Received: 28 February 2024
Revised: 6 April 2024
Accepted: 14 April 2024
Published: 16 April 2024

1. Introduction

As an effective tool to alleviate information overload for users, recommendation systems (RSs) can help users improve their experience in various aspects of life, entertainment, and shopping [1]. However, traditional recommendation methods rely on user personal information or long-term historical records to recommend items. This dependency on available information can lead to poor performance or malfunctioning of RS when such information is unavailable. Session-based recommendation (SBR) [2], as a novel recommendation paradigm, has garnered extensive attention from researchers. SBR operates independently of users' personal information and recommends the next interested item based on the anonymous user's behavior sequence within a session.

Traditional SBR methods, such as collaborative filtering [3] and matrix factorization[4], mostly utilize a matrix to calculate the similarity between items and users, which ignores the recent preferences of users and the temporal correlation of session sequences. The method, which is based on Markov chains, employs the previous interaction item to guide the next interaction item, while the user's preference is not only related to the current behavior, but also has a strong correlation with the past behavior. Recent research based on deep learning techniques has made significant progress, including recurrent neural networks (RNNs) [5], convolutional neural networks (CNNs) [6], and graph neural networks (GNNs) [7,8]. In

RNN-based methods, recommendations are implemented according to the sequence of items, and cannot obtain the information of other sessions. CNN-based algorithms can learn feature representations of items within each session sequence through convolution and pooling operations, but lack the ability to model time series. Methods based on GNNs map interaction items into a graph according to the sequence, utilize the information of other sessions, but struggle to acquire higher graph-level information. The hypergraph structure, which is proposed in DHCN [9], can effectively capture higher graph-level information, while the superior recommendation performance of SBR is obtained through hypergraph neural networks (HGNNs). However, there are still the following defects: firstly, the dependency relationship between the items in the current session and other sessions is not fully considered; secondly, most of the existing SBR models ignore the aggregation of temporal information of items; thirdly, the hypergraph constructed by the session sequence has a serious issue of sparsity.

Based on the above reasons, this paper proposes a spatio-temporal contrastive heterogeneous graph attention network (STC-HGAT), which constructs the category nodes of items through latent Dirichlet allocation (LDA) and considers the dependency relationship between items in the current session and other sessions. Both the spatial and temporal information of the session sequence are considered through the spatial heterogeneous hypergraph and the temporal heterogeneous graph, respectively. The hypergraph attention network (HyperGAT) is used to capture the spatial structure information of the session sequence, and the heterogeneous graph attention network (HGAT) aggregates temporal information of interaction items. To enhance the generalization performance of different session sequences, a modulation factor is added to the loss function. In addition, contrastive learning is used as an auxiliary task to maximize the spatio-temporal mutual information of the sequence.

In summary, the main contributions of this paper are summarized as follows:

- The session sequence is modeled as a spatial heterogeneous hypergraph, and category nodes of items by LDA are introduced to enhance the context information of the hypergraph, which further captures the spatial structure information of sequences.
- The temporal heterogeneous graph aggregates the temporal information of the interaction items, and the features of items effectively fuse the temporal and spatial information of sequences through sumpooling.
- An adaptive weight (AW) loss function is constructed to improve the generalization ability of the STC-HGAT model, and contrastive learning is introduced as an auxiliary task to alleviate the sparsity problem of session data.
- Extensive experiments on real datasets show that the session recommendation performance of our method STC-HGAT outperforms other baseline models.

The rest of this paper is organized as follows: Related work is presented in Section 2. Section 3 provides the problem definition, and introduces the session graph construction method of the STC-HGAT model. In Section 4, the framework and implementation method are described in detail. Extensive experimental results and analysis are presented in Section 5. Finally, in Section 6, we conclude and summarize the entire work.

2. Related Work

In this section, previous work related to the proposed STC-HGAT model is presented. It includes traditional SBR methods, deep learning-based methods, and hypergraph-based methods.

2.1. Traditional SBR Methods

Traditional session-based recommendation methods mainly rely on collaborative filtering, matrix factorization, and Markov chains. In 2011, Park et al. [10] proposed session-based collaborative filtering, in which users prefer the same items in similar sessions. Approaches based on matrix factorization and collaborative filtering do not consider the item sequence. Therefore, FPMC [11] was introduced in 2010, which combines matrix

factorization and Markov chains. It uses Markov chains to learn personalized transition matrices for each user.

However, previous SBR models based on traditional methods rely too much on the sequence order, which leads to a disadvantage when capturing the transition relationship between items over long distances.

2.2. Deep Learning-Based Methods

With the advancement of deep learning, deep learning-based methods have made significant progress in RSs. GRU4Rec [12], introduced in 2015, was the first model to use gated recurrent units to model item interaction sequences. STAMP [13] proposed a short-term attention memory model to capture the current interests of the user. Deep learning-based methods are excellent at capturing the sequential dependencies of session sequences, but ignore the information between interaction items.

In recent years, methods based on GNNs [14,15] have received extensive attention and achieved remarkable results in RSs. GNNs model session sequences as graphs, and aggregate one-hop or multi-hop neighboring item features in the graph to acquire richer representations of items. SR-GNN [16], proposed in 2019, is a pioneering GNN-based method that converts a session into a directed unweighted graph and exploits a gated GNN to generate the session representation. SGNN-HN [17] was proposed in 2020 to introduce a star node for each session graph to capture long-distance relationships between items. While these studies have yielded encouraging results, independent learning from short sessions may be inherently insufficient to accurately reveal a user's true intentions. Research methods are now shifting towards combining collaborative information from other sessions to serve the current session. GCE-GNN [18] exploits the global and local relationships of sessions to enhance the session-level representation.

Although the above methods based on GNNs have achieved good results, these methods only consider the simple transition relationships between items in the session sequence, ignoring the more high-order transition relationships in other sessions from a global perspective.

2.3. Hypergraph-Based Methods

Traditional GNN-based methods [19] primarily leverage GNNs to capture pairwise transition relationship between nodes in graph structure data. Wang et al. introduced hypergraph convolutional networks to capture high-order item relationships within individual sessions and employed self-supervised learning to enhance session representations. In 2022, Li et al. [20] proposed a novel hypergraph-based model HIDE, which models each session as a hypergraph and models the possible interest transfer of users from different perspectives. The application of a hypergraph neural network to SBR can effectively mine the spatial information of the sequence, but still struggles to overcome sparsity issues, especially for short sequences, and these methods do not aggregate the temporal information of interaction items.

We propose a spatio-temporal contrastive heterogeneous graph attention network model, which constructs a spatio-temporal heterogeneous graph to consider both spatial and temporal information of the session sequence, and introduces an LDA algorithm to construct category nodes of items to enhance the context information of the hypergraph. Contrastive learning serves as an auxiliary task to maximize the mutual information between the spatial and temporal representations of sequences, aiming to alleviate the sparsity of data.

3. Preliminary

In this section, the problem of SBR is first defined, and then two constructions of the session graph are introduced.

3.1. Problem Definition

SBR outputs the top K candidate items with higher scores, which enables calculation of its similarity with the session representation. Considering a session dataset $S = \{s_1, s_2, \cdots, s_M\}$ and item datasets $V = \{v_1, v_2, \cdots, v_N\}$, M and N represent the number of S and V, respectively. A given session $s_i = [(v_{s_i,1}, t_1), (v_{s_i,2}, t_2), \cdots, (v_{s_n,1}, t_n)]$ represents the i-th time series of an anonymous user, where $(v_{s_i,k}, t_k) \in V$ denotes the k-th item that the user interacted with in s_i, and t_k is the time of $v_{s_i,k}$ clicked. The purpose of SBR is to predict the next most likely item $\hat{v}_{s_{i,n+1}}$, based on datasets S and V, namely, $\hat{v}_{s_{i,n+1}} = g(S, V, s_i : \theta)$. Here, θ represents the set of learnable parameters and the function $g(\cdot)$ is constructed.

3.2. Session Graph Construction

3.2.1. Spatial Heterogeneous Hypergraph

To explore the spatial structure information of the sequence, we model a single session sequence as a hypergraph $G_s = \{V_s, E_s\}$ in the form of a sliding window, where each sliding window is a hyperedge, and the items in the sliding window are the nodes of the hyperedge. Let e_s^w denote the set of all hyperedges constructed with sliding windows of size w, and then the set of hyperedges of different sizes is gathered together to obtain the set of hyperedges $E_s = e_s^1 \cup e_s^2 \cup \cdots \cup e_s^W$, where W is the maximum size of the sliding window. However, the hypergraph-based recommendation model has a serious issue of graph sparsity. Latent Dirichlet allocation (LDA), as a generative model, can learn the distribution of topics from a set of documents. LDA mines the potential category $c_i = (\theta_1, \theta_2, \cdots, \theta_k)$ from the item dataset V, k represents the number of items in the category, the average features of the top k items within the category are aggregated, and each category feature is denoted as $h_{c_i} = \frac{1}{k} \sum_{j=1}^{k} h_{\theta_j} (\theta_j \in c_i)$. The category nodes are added to the sequence to construct a spatial heterogeneous hypergraph.

3.2.2. Temporal Heterogeneous Graph

The heterogeneous hypergraph mines the spatial information of the sequence, but the temporal information of the interaction items is also important for recommendation. Inspired by DRGN [21], we construct the interaction sequence as a temporal heterogeneous graph $G_t = \{V_t, E_t\}$. For effective training and reduced computation, we aggregate the temporal order of the interactions of the 1-hop neighboring nodes. Taking the sequence s_i as an example, the item $(v_{s_i,k}, t_k) \in V$ is selected as the anchor node to construct the time series $v_{s_i,k} = \{s_1, s_2, \cdots, s_m\}$.

4. Methodology

The framework of the model STC-HGAT is shown in Figure 1, where five modules are introduced as follows: Module I represents HyperGAT to capture higher spatial information between items in the sequence. Module II is HGAT to aggregate the different temporal information of items. Module III fuses the spatial and temporal information of the sequence and obtains the final session representation using the attention mechanism. Module IV denotes contrastive learning (CL). Module V predicts the possible item.

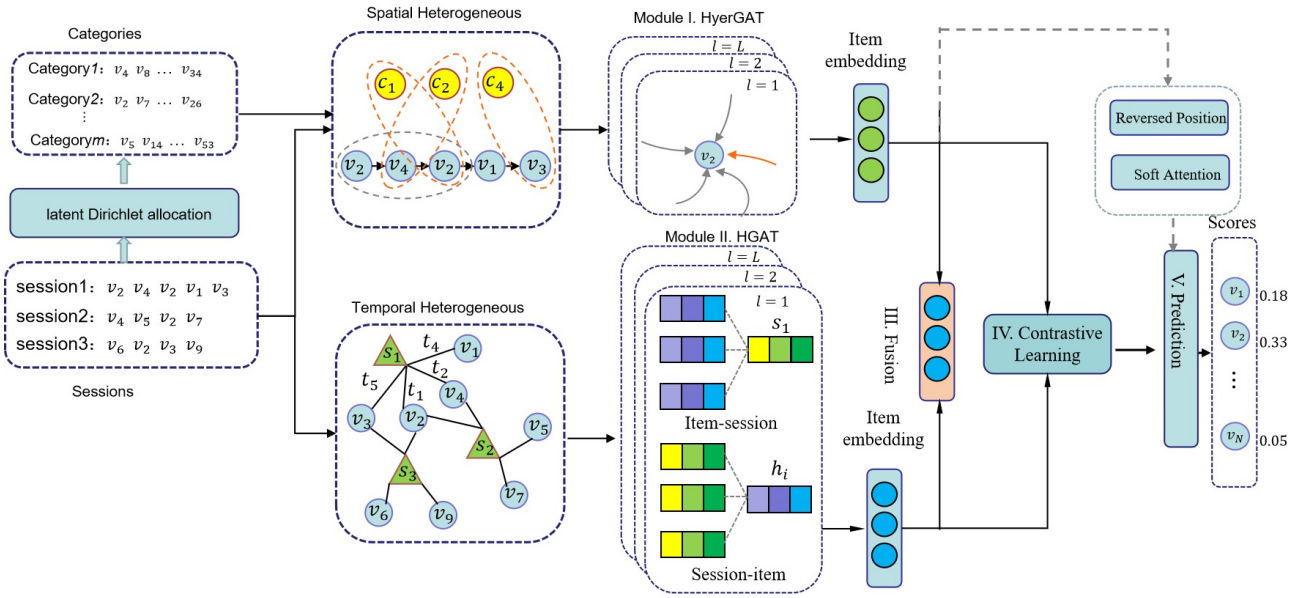

Figure 1. The framework of the model STC-HGAT.

4.1. HyperGAT

After building the session sequence as a spatial heterogeneous hypergraph, we employ HyperGAT as the aggregation operation to learn the importance degree between hyperedges and nodes. The nodes of the item and category in the hypergraph are initially encoded $h^{(0)} = \left\{h_1^{(0)}, h_2^{(0)}, \cdots, h_N^{(0)}, h_c^{(0)}\right\}$, which serves as the input of the first layer.

4.1.1. Nodes to Hyperedges

The information of all the connected nodes N_k to the hyperedge e_k is aggregated via an attention operation, represented as $e_k^{(1)}$:

$$e_k^{(1)} = \sum_{t \in N_k} \alpha_{k,t} W_1^{(1)} h_t^{(0)}, \tag{1}$$

where, $W_1^{(1)}$ is the transformation matrix. $\alpha_{k,t}$ denotes the attention score of the node t on the hyperedge e_k, calculated as follows:

$$\alpha_{k,t} = \frac{S(\hat{W}_1^{(1)} h_t^{(0)}, u^{(1)})}{\sum_{f \in N_k} S(\hat{W}_1^{(1)} h_f^{(0)}, u^{(1)})}, \tag{2}$$

where, $u^{(1)}$ represents the trainable context vector, and $\hat{W}_1^{(1)}$ is the transformation matrix. $S(\cdot, \cdot)$ uses the scaled click attention to compute the similarity computation between the node encoding and the context vector:

$$S(a, b) = \frac{a^T b}{\sqrt{D}}, \tag{3}$$

where, D represents the dimension size, which can be used for normalization when calculating the similarity score.

4.1.2. Hyperedges to Nodes

The attention mechanism is also employed to aggregate information from hyperedges E_t and to distinguish the significance of different hyperedges. The node information is updated as $n_t^{(1)}$:

$$n_t^{(1)} = \sum_{k \in E_t} \alpha_{t,k} W_2^{(1)} e_k^{(1)} \tag{4}$$

$$\alpha_{t,k} = \frac{S(\hat{W}_2^{(1)} e_k{}^{(1)}, W_3^{(1)} n_t^{(1)})}{\sum_{f \in E_t} S(\hat{W}_2^{(1)} e_f{}^{(1)}, W_3^{(1)} n_t^{(1)})}, \tag{5}$$

where, $W_2^{(1)}$, $\hat{W}_2^{(1)}$, and $W_3^{(1)}$ are the trainable weight transformation matrix. $\alpha_{t,k}$ represents the degree of influence of the hyperedge e_k on the node t, $S(\cdot,\cdot)$ as shown in Equation (4). HyperGAT can obtain the information of the direct neighbors, and higher-order information can be learned from the hypergraph through the l HyperGAT layers. The final node representation is denoted as $h^{(l)} = \left\{h_1^{(l)}, h_2^{(l)}, \cdots, h_N^{(l)}, h_c^{(l)}\right\}$.

4.2. HGAT

4.2.1. Items to Session

HGAT aggregates the information of items in the temporal heterogeneous graph. For a given item v_i, the influence degree of the item on the session sequence s is denoted as $e_{i,s}$:

$$e_{i,s} = LeakyReLU(v_{i,s}^T(h_s^{(0)} \odot h_i^{(0)})), \tag{6}$$

where, $v_{i,s}$ presents the weight vector between items and session, and $h_s^{(0)}$ is the encoding of the session, namely, $h_s^{(0)} = \frac{1}{n}\sum_{i=1}^{n} h_i{}^{(0)}$. To ensure comparability between the different nodes, the attention coefficients $\beta_{i,s}$ are normalized by the softmax function on $e_{i,s}$:

$$\beta_{i,s} = \frac{exp(e_{i,s})}{\sum_{v_i \in s} exp(e_{i,s})} \tag{7}$$

Then, the feature representation of the session $h_s^{(1)}$ is obtained:

$$h_s^{(1)} = \sum_{v_i \in s} \beta_{i,s} h_i^{(0)} \tag{8}$$

4.2.2. Sessions to Item

Similarly, the attention coefficient $\beta_{s,i}$ of the sequence s to v_i is obtained through the LeakyReLU and softmax function:

$$e_{s,i} = LeakyReLU(v_{s,i}^T(h_i^{(0)} \odot h_s^{(1)})) \tag{9}$$

$$\beta_{s,i} = \frac{exp(e_{s,i})}{\sum_{s \in v_{s_i,k}} exp(e_{s,i})}. \tag{10}$$

Then, the feature representation of the item is acquired:

$$h_{t,i}^{(1)} = \sum_{s \in v_{s_i,k}} \beta_{s,i} h_s^{(1)} \tag{11}$$

After d layers of HGAT, the final node features are denoted by $h_t^{(d)} = \left\{h_1^{(d)}, h_2^{(d)}, \cdots, h_N^{(d)}\right\}$.

4.3. Information Fusion Layer

The temporal and spatial information of the sequence are fused by sumpooling; h represents the aggregated session vector:

$$h = sumpooling(h_s^{(l)} + h_t^{(d)}) \tag{12}$$

Position information, as an effective technique, is introduced in Transformer to memorize the position information of items. We reverse the position encoding $P = [p_1, p_2, \cdots, p_n]$, where p_i represents the encoding vector for the i-th position, and n is the length of the current session sequence. In the current session, the encoding information h_i^* for the i-th item is obtained through aggregation operations and non-linear transformations:

$$h_i^* = tanh(W_4(h_i||p_{n-i+1}) + b) \tag{13}$$

$$\rho_i = p^T(W_5 h_i^* + W_6 h_i + c) \tag{14}$$

$$s = \sum_{i=1}^{l} \rho_i \cdot h_i \tag{15}$$

in which, W_4, W_5, W_6, and b, c are trainable parameters. The session sequence s is represented via the soft attention mechanism.

4.4. Contrastive Learning

Comparing the spatio-temporal representations of items, if the two item embedding representations are the same, the pair of embeddings is marked as ground-truth, otherwise it is marked as negative. The model adopts InfoNCE, employing the standard binary cross-entropy loss between positive samples and negative samples as the learning objective, denoted as:

$$\mathcal{L}_c = -log\sigma(f_D(h_i^H, h_i^I)) - log\sigma(1 - f_D(\tilde{h}_i^H, h_i^I)), \tag{16}$$

where, $\tilde{h}_i^H$ represents negative samples, which are obtained by corrupting h_i^H (row transformation and column transformation). The discriminant function $f_D(.,.)$ conducts the dot product of the vectors to evaluate the consistency between the two input vectors.

4.5. Prediction Layer

The dot product between the current session representation s and the item representation h_i is calculated to the likelihood of the recommended candidate item, and normalized by the softmax function to obtain the probability of the next clicked item $\hat{y}_i$:

$$\hat{y}_i = \text{softmax}(s^T h_i) \tag{17}$$

The differences between samples within the dataset also reflect variations in the model's predictions for different sessions [22]. We add a modulation factor to the cross-entropy loss function to allocate weights based on the predictive deviation of samples, thereby constructing the adaptive weight (AW) loss function:

$$p_i = \begin{cases} \hat{y}_i, & if\ y_i = 1 \\ 1 - \hat{y}_i, & otherwise \end{cases} \tag{18}$$

$$\mathcal{L}_r(\hat{y}_i) = -\sum_{i=1}^{N} (2 - 2p_i)^{\gamma} \log(1 - p_i) \tag{19}$$

in which, γ is the temperature coefficient, and y_i represents the one-hot encoding vector of the ground-truth item. Ultimately, the model unifies the objective function of the

recommendation task and the function of contrastive learning, where the former is the main task of recommendation, and the latter is the auxiliary task, specifically defined as:

$$\mathcal{L} = \mathcal{L}_r + \lambda \mathcal{L}_c, \tag{20}$$

where λ is used to control the size of the contrastive learning task.

5. Experiment and Analysis

In this section, we introduce the datasets and preprocessing methods, as well as the evaluation metrics, which is followed by the baselines, and the parameter settings of the STC-HGAT model. Finally, we conduct extensive experiments to address the following questions:

- RQ1: Is the STC-HGAT model superior to state-of-the-art baseline models?
- RQ2: How do different modules contribute to the recommendation performance in the STC-HGAT model?
- RQ3: What impact do different hyperparameter settings have on the final recommendation results?
- RQ4: How does the STC-HGAT model perform when dealing with sessions of different lengths?
- RQ5: How about the computational complexity of the STC-HGAT model, compared with other models?

5.1. Experimental Settings

5.1.1. Datasets

This work selects two real datasets, namely, Diginetica and Yoochoose. The Diginetica dataset is from the CIKM Cup 2016 and contains transaction data of anonymous users. Yoochoose was released in the Recsys Challenge 2015 as a public dataset, which mainly consists of click data from an e-commerce website within six months. Yoochoose1/4 and Yoochoose1/64 are both from the public dataset Yoochoose, which was released in the RecSys Challenge. While the Yoochoose dataset consists of the huge interaction data of users, only part of them are usually selected as the dataset of SBR. The Yoochoose1/4 dataset is the training set with the most recent quarter of the interactions in the Yoochoose dataset. Yoochoose1/64 is different in that it takes 1/64 of the most recent interactions as the training set.

For better recommendation, we follow [23,24] and preprocess the three datasets. In the preprocessing phase, sessions of length l with less than five occurrences of items were filtered out on both datasets. For the Yoochoose dataset, the sessions that occurred on the last day were used for the test set and the rest were used as the training set. The difference is that for the Diginetica dataset, the sessions that occurred within the last seven days are used for the test set. Meanwhile, the session $s_i = \{[(v_{s_i,1}, t_1), (v_{s_i,2}, t_2), \cdots, (v_{s_i,n}, t_n)\}$ is preprocessed by sequence segmentation to generate sequences and corresponding labels, such as $\{[(v_{s_i,1}, t_1)], (v_{s_i,2}, t_2)\}, \cdots, \{[(v_{s_i,1}, t_1), (v_{s_i,2}, t_2), \cdots, (v_{s_n,1}, t_{n-1})], (v_{s_n,1}, t_n)\}$, which are used for training and testing on the three datasets. Table 1 shows the detailed statistics of the dataset after preprocessing.

Table 1. Statistical information of the datasets.

Dataset	Diginetica	Yoochoose1/64	Yoochoose1/4
#train	719,470	369,859	5,917,746
#test	60,858	55,898	55,898
#items	43,596	16,766	29,618
avg.len	5.12	6.16	5.71

5.1.2. Metrics

To facilitate the analysis and comparison, we follow previous work and choose the evaluation metrics widely used in SBR, including *P*@20 and *MRR*@20.

P@K (precision calculated over the top *K* items) is widely employed to measure the accuracy of a prediction. It represents the fraction of correctly recommended items among the top K items, and the equation is:

$$P@K = \frac{n_{hit}}{K}, \tag{21}$$

where n_{hit} is the number of correctly recommended items in the top K-ranked list.

MRR@K (mean reciprocal rank calculated over the top K items) is the mean reciprocal rank calculated over the top K items, which takes into account the order in which the recommendations are ranked. The *MRR@K* value increases as the accuracy of the top-ranked items recommended by the model. Formally, *MRR@K* is calculated as follows:

$$MRR@K = \frac{1}{|S|} \sum_{x=1}^{S} \frac{1}{\text{rank}_x}, \tag{22}$$

in which, *S* is the number of items in the test set, and $rank_x$ is the user interaction in the x position actual list of recommended items. In the experiments, the value of *K* is set to 20; that is, we investigate the values of *MRR*@20 and *P*@20 of different models.

5.1.3. Baselines

Our proposed model is compared with related session recommendation models to verify the advantages of STC-HGAT. The selected baseline models are as follows:

1. Traditional method: FPMC combines matrix decomposition and first-order MCs to capture user preferences. Similar to previous studies of SBR [25], we also ignore the potential representation of the user when calculating the recommendation score.
2. Attention mechanism and RNNs: GRU4REC learns the final sequence representation through gated neural networks. STAMP replaces RNNs in previous work with multiple attention layers, and captures the short-term interest of the user through the last item in the current session.
3. GNNs: SR-GNN constructs the session sequence as a graph, and uses gated GNNs to acquire the encoded representation of items. GCE-GNN considers other session information to construct a global graph.
4. HGNNs: SHARE [26] builds each session as a hypergraph that models item correlations defined by various context windows. HIDE employs the hyperedges of the current session and captures user interest shifts from different perspectives.
5. Contrastive learning: COTREC [27] utilizes different connectivity information to generate labels and supervise each other through contrastive learning. HCCF [28] acquires the global relationship through the hypergraph and the local neighborhood through the original interaction graph, and designs the enhancement method of double graph contrastive learning.
6. Temporal: STAM [29] uses the multi-head attention mechanism to explore the relationship in the sequence and aggregate the spatio-temporal information of the neighbor node embedding. TMI-GNN [30] identifies multiple interests of modeling users through temporal information of the interval and interest level of item transitions.

5.1.4. Hyperparameter Settings

Adaptive Moment Estimation (Adam) refers to an optimization algorithm used in training neural networks, particularly in SBR models. In this experiment, Adam was used as the optimizer and the learning rate was set to 0.001 for the datasets. For a fair comparison, the embedding size of the STC-HGAT and baseline models is set to 128. For the Yoochoose1/4 dataset, we set the number of HyperGAT to 2. Let us set the number of

layers to 3. As in previous work, 10% was randomly sampled from the training set as the validation set for parameter tuning.

5.2. Overall Performance Comparison (RQ1)

As shown in Table 2, the experimental results of the STC-HGAT and baseline models are compared to illustrate the overall performance of the proposed model, and we average the values of results in the table after multiple experiments. The overall performance of STC-HGAT achieves the optimal outcome. Multiple GRU layers are simply stacked, and the order relationship is only considered in GRU4Rec. GRU4Rec models the session sequence as a session representation, and although its performance is better than FPMC, it is not sufficient to consider only modeling the sequence because user preferences are dynamic. The attention mechanism is employed in STAMP, which can consider the weight difference between different items of the session to express the user's intention more accurately. This indicates that assigning different attention weights to different items for session encoding is very effective. Therefore, the STC-HGAT model proposed in this paper uses HyperGAT to achieve the spatial information of the sequence, and HGAT to learn the temporal information of the item sequence.

Table 2. Performance comparison of our STC-HGAT model with the baselines.

Methods	Diginetica		Yoochoose1/64		Yoochoose1/4	
	P@20	**MRR@20**	**P@20**	**MRR@20**	**P@20**	**MRR@20**
FPMC	26.31	7.63	55.63	19.67	40.78	17.41
GRU4Rec	30.79	8.22	60.84	22.89	43.80	19.83
STAMP	48.32	16.00	68.35	28.63	54.17	26.11
SR-GNN	46.62	15.13	67.85	27.71	53.96	25.46
GCE-GNN	51.26	17.78	70.57	30.94	58.87	26.47
SHARE	54.22	18.54	71.31	30.58	63.75	27.29
HIDE	<u>54.73</u>	<u>18.85</u>	<u>71.53</u>	<u>31.24</u>	71.59	30.13
COTREC	53.66	18.51	70.53	30.12	54.91	27.82
HCCF	54.36	18.67	71.46	30.95	<u>71.86</u>	<u>30.99</u>
STAM	53.18	18.44	70.86	29.80	70.17	29.97
TMI-GNN	54.13	18.61	71.22	30.81	70.76	29.22
STC-HGAT	**55.40**	**19.01**	**72.12**	**31.57**	**72.62**	**31.19**
p value	0.001	0.001	0.001	0.01	0.001	0.001

Note: The black fonts and underlined fonts indicate the best results for each column and baselines respectively.

Compared with the GNN-based models, including SR-GNN and GCE-GNN, the recommendation effect is better than that of the deep learning-based models. This is because the session sequence is modeled as a graph structure, while the GNN-based models employ GNNs to learn the pairwise transition relationship between items, which also proves the importance of considering the dependencies between items when recommending. The performance of DHCN and SHARE is better than that of the GNN model. The GNN-based model only constructs the session as a simple graph construction, which cannot capture complex high-order relationships between items, while hypergraph further mines the spatial information of the session sequences. However, the method based on contrastive learning alleviates the data sparsity of the graph model and shows better recommendation performance. Different from the above baseline models, the category nodes of items are considered through the LDA algorithm, so it can effectively represent the characteristics of the item in the current session.

Although models based on time series information, including STAM and TMI-GNN, mine the temporal information in the sequence, they produce lower results than the contrastive learning model, which may be due to the influence of sparsity and sample imbalance. In our STC-HGAT approach, the heterogeneous graph and hypergraph are employed to simultaneously consider the spatio-temporal information of sequences, and contrastive learning is added in the

prediction to alleviate the data sparsity. This is also the reason for the superior performance of STC-HGAT. For both the evaluation metrics, *MRR*@20 and *P*@20, the *p*-value represents the significant improvement of STC-HGAT over the best baseline, and the *p*-values of STC-HGAT are all less than the critical value of 0.05.

5.3. Ablation Studies (RQ2)

To verify the performance effect of different modules in the STC-HGAT model, we use the following variants for evaluation:

- STC-HGAT-S removes the spatial heterogeneous hypergraph and only retains the temporal information of the temporal heterogeneous graph.
- STC-HGAT-T extracts the sequence information through the spatial heterogeneous hypergraph, and then calculates the score of the candidate.
- STC-HGAT-C removes the category nodes and only aggregates the hyperedge information in the sliding window.
- ST-HGAT does not employ contrastive learning as an auxiliary task.
- STC-HGAT-A only employs the cross-entropy function after removing the modulation factor.

We can see from Table 3 that the proposed STC-HGAT achieves the best performance on *P*@20 and *MRR*@20. On the Diginetica, Yoochoose1/64, and Yoochoose1/4 datasets, the STC-HGAT model outperforms the STC-HGAT-S variant, which shows that modeling the session sequence as a spatial heterogeneous hypergraph and using HyperGAT can effectively learn the spatial information from the sequence. Meanwhile, the STC-HGAT model performs better than the STC-HGAT-T variant, which proves the effectiveness of using HGAT to learn temporal information. By comparing the experimental results of the STC-HGAT model and the ST-HGAT variant, it can be seen that the use of CL can effectively enhance the graph model modeling, and alleviate the sparsity problem of graph data to a certain extent. The performance of STC-HGAT is better than STC-HGAT-C, and LDA enriches the contextual semantic information of the graph structure data. However, the STC-HGAT-C variant shows the best performance on the Yoochoose1/64 dataset and the corresponding *MRR*@20, which may be because the session length in the Yoochoose1/64 dataset is relatively long, and the model does not fully consider the position information of the long session sequence when the category nodes are added. On the whole, STC-HGAT-A performs slightly worse than STC-HGAT on the three datasets, indicating the effectiveness of the AW loss function. Ablation experiments show that the fusion of temporal and spatial information of the sequence can effectively learn the item feature representation, and the introduction of contrastive learning can enhance graph modeling to a certain extent, thereby alleviating the sparsity of data. And adding an adaptive loss function can improve the generalization ability of the model.

Table 3. Experimental results of different modules.

Methods	Diginetica		Yoochoose1/64		Yoochoose1/4	
	P@20	**MRR@20**	**P@20**	**MRR@20**	**P@20**	**MRR@20**
STC-HGAT-S	54.35	18.27	71.23	30.85	71.13	29.24
STC-HGAT-T	51.13	17.38	69.54	29.63	68.20	27.50
STC-HGAT-C	53.86	18.13	70.31	**31.65**	70.03	28.72
ST-HGAT	55.00	18.59	71.55	31.18	70.67	29.49
STC-HGAT-A	55.12	18.78	71.86	31.44	71.87	30.67
STC-HGAT	**55.40**	**19.01**	**72.19**	31.57	**72.62**	**31.19**

Note: The black fonts indicate the best results for each column.

5.4. Impact of Hyperparameters (RQ3)

5.4.1. Click Unit

On the two evaluation metrics, the values of the click unit effects on the recommendation performance are different when the session sequence is modeled as a hypergraph. Here, the value of the click unit is 1, which means that the session sequence has not been built into a hypergraph, while the item uses static embeddings throughout the session. From Figure 2, it can be observed that on the Diginetica dataset, the recommendation performance is optimal when the corresponding click units are 5 and 1 for *P*@20 and *MRR*@20, respectively. However, in the Yoochoose1/64 dataset and the Yoochoose1/4 dataset, the recommendation performance of *P*@20 is optimal when the click unit value is set to 7 and 6, while for the *MRR*@20 evaluation metric, the click unit value of the Yoochoose dataset is 2 to achieve the optimal outcome. This shows that for *P*@20, it is necessary to set a large click unit value to effectively acquire the feature information of items, so as to improve the hit rate of recommendation. For *MRR*@20, we need to consider the location accuracy of the item for recommendation. If the click unit is set too high, it is easy to take irrelevant item information into account when establishing the hypergraph. Therefore, we need to set the click unit to a small value. The results show that, in some cases, making a trade-off between the hit rate and the item ranking is important when choosing the click unit.

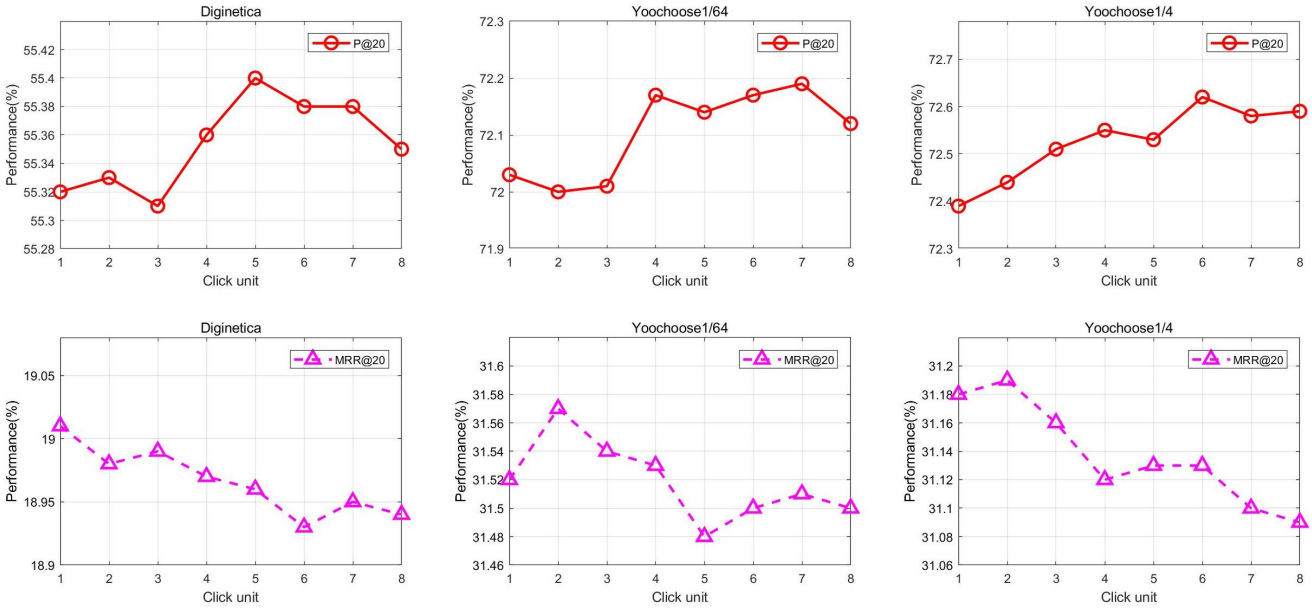

Figure 2. The impact of the click unit on recommendation performance.

5.4.2. CL Parameter λ

Figure 3 illustrates the impact of the parameter controlling the size of the auxiliary task for CL on the experimental results. Specifically, experiments were undertaken to explore the value of the typical values set for 0.001 to 0.01. It can be seen from the figure that different values also have different effects on the recommendation performance, and the recommendation performance decreases with larger values, which may be caused by the conflict between the two gradients of the recommendation task and the self-supervised learning task. On the Diginetica dataset and the two evaluation metrics, the recommendation effect is best when the value is set to 0.001, which indicates that for the Diginetica dataset and the value is 0.001, the hypergraph channel and the general conversation graph channel can learn more mutual information from each other through CL. For the Yoochoose1/64 dataset and the Yoochoose1/4 dataset, the recommendation effect is best on the corresponding two evaluation indicators when the value is 0.0005. When the recommendation performance is optimal, the value corresponding to the Yoochoose1/64 dataset is relatively small compared

to the Diginetica dataset, which indicates that appropriate values need to be set for different datasets to balance the two different learning tasks.

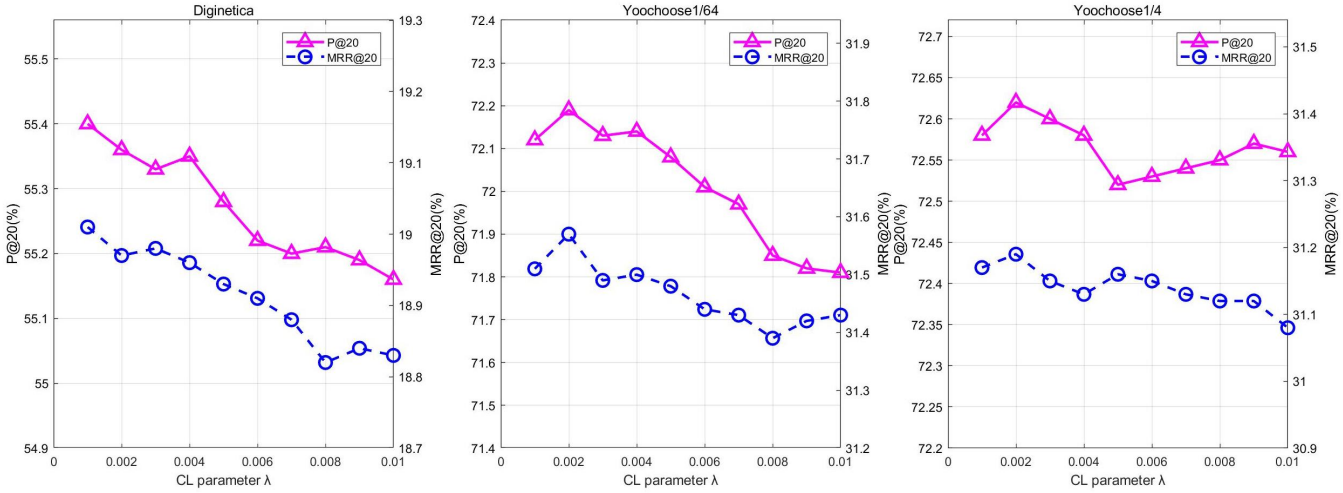

Figure 3. The impact of the CL parameter on recommendation performance.

5.5. Impact of Session Length (RQ4)

For a more thorough analysis, especially when dealing with different data volumes and operating conditions. We investigate the recommendation performance of three different recommendation models, including STC-HGAT, HIDE, and HCCF, when the session length is different. It can be seen from Figure 4 that on the three datasets, the STC-HGAT model reaches the optimal on *P*@20 when the session length is 4, 5, and 4, respectively. However, with increase in the length, the performance of STC-HGAT declines, potentially exceeding the average length of the session sequence in the dataset, and the long sequence is more susceptible to data sparsity. The experimental results show that the performance of STC-HGAT is consistent for different length session sequences.

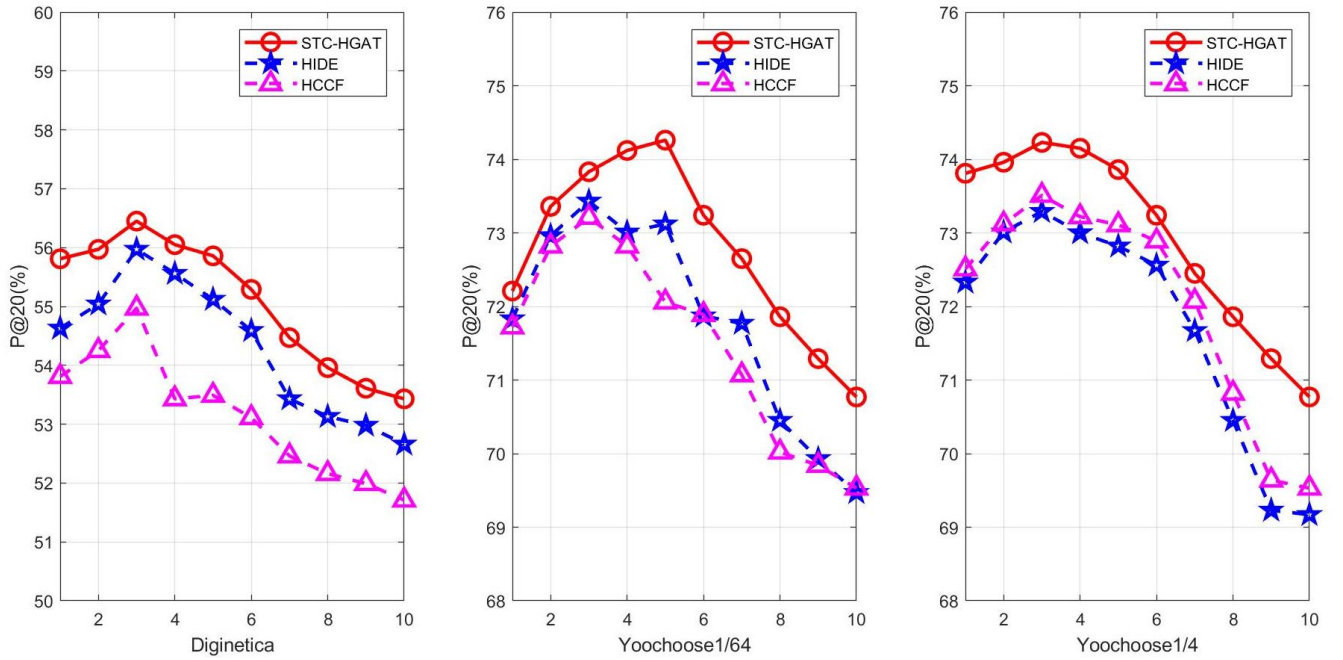

Figure 4. The impact of session length on recommendation performance.

5.6. The Computational Complexity of STC-HGAT (RQ5)

The construction of a spatio-temporal heterogeneous graph may increase the complexity of the model. In order to understand the effectiveness and feasibility of our model in practical applications, we increased the comparison of the training time of each epoch and the number of trainable parameters on three datasets, and compared with three types of recommendation models, namely HIDE, COTREC, and TMI-GNN, respectively. From the results in Table 4, we can see that the parameter values of our model are slightly lower than those of TMI-GNN and HIDE. Although the number of parameters of our model is higher than that of COTREC, its running time is less than that of COTREC because self-supervised learning requires more corrosion operations and runs take more time. The experimental results show that the construction of a spatio-temporal heterogeneous graph does not greatly increase the number of parameters and running time, and STC-HGAT is quite practical to apply.

Table 4. The computational complexity of STC-HGAT.

Methods	Diginetica		Yoochoose1/64		Yoochoose1/4	
	Time	**#Params**	**Time**	**#Params**	**Time**	**#Params**
HIDE	29 m 21 s	4.31 M	18 m 49 s	3.87 M	57 m 69 s	4.12 M
COTREC	33 m 41 s	**4.24 M**	20 m 13 s	3.81 M	63 m 42 s	4.10 M
TMI-GNN	29 m 44 s	4.30 M	19 m 12 s	3.84 M	58 m 93 s	4.14 M
STC-HGAT	**28 m 56 s**	4.26 M	**18 m 33 s**	**3.73 M**	**56 m 33 s**	**4.06 M**

Note: The black fonts indicate the best results for each column, additionally, m, s, and M stand for minute, second, and million, respectively.

6. Conclusions and Discussion

This work investigates the problem of SBR using spatio-temporal heterogeneous graph attention networks (STC-HGAT). We construct session sequences into a spatial heterogeneous hypergraph and a temporal heterogeneous graph, respectively. STC-HGAT uses LDA to construct category nodes, mines spatial information through HyperGAT, and uses HGAT to aggregate temporal information. Then, information from two sessions is fused using a sumpooling operation, and the spatial information and temporal information of the sequence are maximized through CL. A modulation factor was added to the cross-entropy loss function to construct the AW loss function. Experiments undertaken show that the proposed method STC-HGAT exhibits superior performance on the three datasets.

Although the recommendation performance of the STC-HGAT model has been improved, the real-time performance of the recommendation results may need to be considered. In the future, graph compression techniques, such as low-rank factorization, can be used to compress the original large-scale graph data into smaller matrices, thereby reducing the computational complexity. In addition, the noise problem can also affect the recommendation performance of the model, so, sequence time information and item interaction frequency will be further used to remove the noise problem.

Author Contributions: F.Y.: conceptualization, methodology, formal analysis and investigation, writing—original draft preparation; D.P.: conceptualization, methodology, formal analysis and investigation, writing—review and editing, and supervision. All authors have read and agreed to the published version of the manuscript.

Funding: The work is supported by the National Natural Science Foundation of China under Grant No. 61772342.

Data Availability Statement: The authors declare that the data supporting the findings of this study are available within the article.

Acknowledgments: We would like to express our special thanks to the members of our lab for valuable discussions on this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, S.; Cao, L.; Wang, Y.; Sheng, Q.Z.; Orgun, M.A.; Lian, D. A Survey on Session-based Recommender Systems. *ACM Comput. Surv.* **2022**, *54*, 154:1–154:38. [CrossRef]
2. Yang, F.; Peng, D.; Zhang, S. Self-supervised hypergraph neural network for session-based recommendation supported by user continuous topic intent. *Appl. Soft Comput.* **2024**, *154*, 111406. [CrossRef]
3. Papadakis, H.; Papagrigoriou, A.; Panagiotakis, C.; Kosmas, E.; Fragopoulou, P. Collaborative filtering recommender systems taxonomy. *Knowl. Inf. Syst.* **2022**, *64*, 35–74. [CrossRef]
4. Yu, Y.; Gao, Y.; Wang, H.; Wang, R. Joint user knowledge and matrix factorization for recommender systems. *World Wide Web* **2018**, *21*, 1141–1163. [CrossRef]
5. Cui, Q.; Wu, S.; Liu, Q.; Zhong, W.; Wang, L. MV-RNN: A Multi-View Recurrent Neural Network for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* **2020**, *32*, 317–331. [CrossRef]
6. Zhou, X.; Li, Y.; Liang, W. CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support. *IEEE ACM Trans. Comput. Biol. Bioinform.* **2021**, *18*, 912–921. [CrossRef] [PubMed]
7. Wang, Y.; Liu, Z.; Fan, Z.; Sun, L.; Yu, P.S. DSKReG: Differentiable Sampling on Knowledge Graph for Recommendation with Relational GNN. In Proceedings of the CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, 1–5 November 2021; ACM: New York, NY, USA, 2021; pp. 3513–3517.
8. Wan, Z.; Liu, X.; Wang, B.; Qiu, J.; Li, B.; Guo, T.; Chen, G.; Wang, Y. Spatio-temporal Contrastive Learning-enhanced GNNs for Session-based Recommendation. *ACM Trans. Inf. Syst.* **2024**, *42*, 58:1–58:26.
9. Xia, X.; Yin, H.; Yu, J.; Wang, Q.; Cui, L.; Zhang, X. Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation. AAAI Press: Washington, DC, USA, 2021; pp. 4503–4511.
10. Park, S.E.; Lee, S.; Lee, S.g. Session-Based Collaborative Filtering for Predicting the Next Song. In Proceedings of the 2011 First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering, Jeju, Republic of Korea, 23–25 May 2011; pp. 353–358.
11. Rendle, S.; Freudenthaler, C.; Schmidt-Thieme, L. Factorizing personalized Markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, Raleigh, NC, USA, 26–30 April 2010; pp. 811–820.
12. Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; Tikk, D. Session-based Recommendations with Recurrent Neural Networks. In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, 2–4 May 2016.
13. Liu, Q.; Zeng, Y.; Mokhosi, R.; Zhang, H. STAMP: Short-Term Attention/Memory Priority Model for Session-based Recommendation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, 19–23 August 2018; pp. 1831–1839.
14. Guo, S.; Bai, T.; Deng, W. Targeted Shilling Attacks on GNN-based Recommender Systems. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, CIKM 2023, Birmingham, UK, 21–25 October 2023; ACM: New York, NY, USA, 2023; pp. 649–658.
15. Zhang, D.; Zhu, Y.; Dong, Y.; Wang, Y.; Feng, W.; Kharlamov, E.; Tang, J. ApeGNN: Node-Wise Adaptive Aggregation in GNNs for Recommendation. In Proceedings of the Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April–4 May 2023; ACM: New York, NY, USA, 2023; pp. 759–769.
16. Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; Tan, T. *Session-Based Recommendation with Graph Neural Networks*; AAAI Press: Washington, DC, USA, 2019; pp. 346–353.
17. Pan, Z.; Cai, F.; Chen, W.; Chen, H.; de Rijke, M. *Star Graph Neural Networks for Session-Based Recommendation*; ACM: New York, NY, USA, 2020; pp. 1195–1204.
18. Wang, Z.; Wei, W.; Cong, G.; Li, X.; Mao, X.; Qiu, M. *Global Context Enhanced Graph Neural Networks for Session-Based Recommendation*; ACM: New York, NY, USA, 2020; pp. 169–178.
19. Jin, D.; Wang, L.; Zheng, Y.; Song, G.; Jiang, F.; Li, X.; Lin, W.; Pan, S. Dual Intent Enhanced Graph Neural Network for Session-based New Item Recommendation. In Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April–4 May 2023; ACM: New York, NY, USA, 2023; pp. 684–693.
20. Li, Y.; Gao, C.; Luo, H.; Jin, D.; Li, Y. *Enhancing Hypergraph Neural Networks with Intent Disentanglement for Session-Based Recommendation*; ACM: New York, NY, USA, 2022; pp. 1997–2002.
21. Zhang, M.; Wu, S.; Yu, X.; Liu, Q.; Wang, L. Dynamic Graph Neural Networks for Sequential Recommendation. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 4741–4753. [CrossRef]
22. Ouyang, K.; Xu, X.; Chen, M.; Xie, Z.; Zheng, H.; Song, S.; Zhao, Y. *Mining Interest Trends and Adaptively Assigning Sample Weight for Session-Based Recommendation*; ACM: New York, NY, USA, 2023; pp. 2174–2178.
23. Zhang, P.; Guo, J.; Li, C.; Xie, Y.; Kim, J.; Zhang, Y.; Xie, X.; Wang, H.; Kim, S. *Efficiently Leveraging Multi-Level User Intent for Session-Based Recommendation via Atten-Mixer Network*; ACM: New York, NY, USA, 2023; pp. 168–176.
24. Zeyu, H.; Yan, L.; Feng, W.; Wei, Z.; Alenezi, F.; Tiwari, P. Causal embedding of user interest and conformity for long-tail session-based recommendations. *Inf. Sci.* **2023**, *644*, 119167. [CrossRef]
25. Peng, D.; Zhang, S. GC–HGNN: A global-context supported hypergraph neural network for enhancing session-based recommendation. *Electron. Commer. Res. Appl.* **2022**, *52*, 101129. [CrossRef]
26. Wang, J.; Ding, K.; Zhu, Z.; Caverlee, J. *Session-Based Recommendation with Hypergraph Attention Networks*; SIAM: Philadelphia, PA, USA, 2021; pp. 82–90.

27. Xia, X.; Yin, H.; Yu, J.; Shao, Y.; Cui, L. *Self-Supervised Graph Co-Training for Session-Based Recommendation*; ACM: New York, NY, USA, 2021; pp. 2180–2190.
28. Xia, L.; Huang, C.; Xu, Y.; Zhao, J.; Yin, D.; Huang, J.X. *Hypergraph Contrastive Collaborative Filtering*; ACM: New York, NY, USA, 2022; pp. 70–79.
29. Yang, Z.; Ding, M.; Xu, B.; Yang, H.; Tang, J. *STAM: A Spatiotemporal Aggregation Method for Graph Neural Network-Based Recommendation*; ACM: New York, NY, USA, 2022; pp. 3217–3228.
30. Shen, Q.; Zhu, S.; Pang, Y.; Zhang, Y.; Wei, Z. Temporal Aware Multi-Interest Graph Neural Network for Session-Based Recommendation. In Proceedings of the 14th Asian Conference on Machine Learning, Hyderabad, India, 12–14 December 2022; Volume 189.