ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230592

Rethinking Temporal Information in Session-Based Recommendation: A Position-Agnostic Approach

Xianghong Xu[a], Kai Ouyang[a;*], Jiaxin Zou[a], Hai-Tao Zheng[a,b;], Wenqiang Liu[c], Dongxiao Huang[c] and Bei Wu[c]**

[a]Shenzhen International Graduate School, Tsinghua University
[b]Pengcheng Laboratory, 518055, Shenzhen
[c]Interactive Entertainment Group, Tencent Inc.
ORCiD ID: Xianghong Xu https://orcid.org/0000-0003-2447-4107,
Kai Ouyang https://orcid.org/0000-0002-0884-529X, Jiaxin Zou https://orcid.org/0009-0009-7870-0174,
Hai-Tao Zheng https://orcid.org/0000-0001-5128-5649, Wenqiang Liu https://orcid.org/0000-0003-4577-407X,
Dongxiao Huang https://orcid.org/0009-0002-8673-6504, Bei Wu https://orcid.org/0009-0003-8653-3835

Abstract. Session-based Recommendation (SBR) aims to predict the next item for a session, which consists of several clicked items in a transaction. Most SBR approaches follow an underlying assumption that all sequential information should be strictly utilized. Thus, they model temporal information for items using implicit, explicit, or ensemble methods. In fact, users may recall previously clicked items but might not remember the exact order in which they were clicked. Therefore, focusing on representing item temporal information in various ways could make learning session intents challenging. In this paper, we rethink the necessity of temporal information for items in SBR. We propose Aggregating the Contextual intents of the session with Attentive networks, namely ACARec. Specifically, we avoid explicitly modeling positional embeddings and learn contextual intents through aggregation methods (convolutions or poolings). We also demonstrate that even an entirely position-agnostic aggregation approach can yield promising results. Extensive experiments on real-world datasets validate our arguments. We hope our study can provide insights into SBR and inspire future research in the community.

1 Introduction

Session-based Recommendation (SBR) has attracted considerable attention in scenarios where long-term user profiles are unavailable or users are not logged in. A session consists of a list of clicked items in a transaction, ordered by their timestamps. SBR aims to predict the next item for the current anonymous session [27]. The primary challenge of SBR lies in effectively utilizing the limited information to make recommendations with sufficient accuracy.

Existing approaches to SBR can be categorized as traditional methods and deep neural network (DNN) models. The latter can be further divided into implicit, explicit, and ensemble categories based on the strategies they employ for temporal (i.e., positional) modeling. Representative traditional methods can be categorized into four classes: rule-based [22], K Nearest Neighbor (KNN) based [10], Markov Chain (MC) based [19], and probabilistic based [42]. These early efforts are relatively simple but their performance is compromised by the data sparsity issue [2, 14] and they are frequired to maintain numerous parameters [43].

With the boom of deep learning, recurrent neural networks (RNNs) have exhibited overwhelming advantages in modeling sequential data, and they have been introduced into SBR to capture sequential order between items and bring impressive performance [3, 4, 23]. These models make an underlying assumption that all sequential information of items should be strictly utilized. treat sessions as unidirectional sequences. Subsequently, some studies attempted to utilize the attention mechanism to capture the main purpose of sessions. Besides, the convolutional neural network (CNN) is also introduced to capture temporal relations between items. Therefore, positional ID embeddings are introduced to enhance item representations [7, 8, 5, 24, 41]. Afterward, some studies [29, 17, 28] have discovered that sessions are not akin to natural languages or strictly ordered sequences. Therefore, using RNN might lead the model to overlook the coherence within sessions. For instance, listening to an album in random order or in sequence will generate two different sessions, but the anonymous user may have similar intents in both cases. Therefore, they attempt to transform sessions into directed graphs to represent the relative orders of items, where items constitute the nodes and the directed edges represent the order in which they were clicked. As a result, Graph Neural Networks (GNN) [31, 32] are thoroughly explored in SBR. We observe that both RNN and directed graphs are structurally capable of representing the positional information in an **implicit** way. Utilizing positional ID embeddings for items is an **explicit** approach to modeling item positions.

Thereafter, it is natural to attempt to combine both implicit and explicit approaches to model the temporal information, namely **ensemble** methods. For example, combining RNN and the attention mechanism with positional ID embeddings [7]. Moreover, stacking GNN layers and attention layers with position embedding is also explored. In recent years, ensemble methods have been gaining increasing popularity and attention in SBR and some advanced techniques are introduced [35, 33].

Notwithstanding that implicit, explicit, and ensemble models have

[*] Equal contribution.
[**] Corresponding author. Email: zheng.haitao@sz.tsinghua.edu.cn

made much progress, the underlying assumption they followed may limit the performance. In fact, users may recall previously clicked items when they make decisions in a session, but might not remember the exact order in which they were clicked. Consequently, the assumption may bring a worse inductive bias [11] in SBR. Therefore, in this paper, we revisit the evolution of the three categories of SBR models and pose a fundamental research question: **is it really necessary to model the exact positional information or relative orders of items in sessions?**

To address these issues and validate our argument, we propose a concise position-agnostic model with well-known techniques. To this end, we **A**ggregate **C**ontextual intents of sessions with **A**ttentive networks for SBR, namely **ACARec**. The most crucial difference between ACARec and the previous works is ACARec attempts to utilize an entirely position-agnostic approach to learn session intents. Specifically, ACARec does not have positional ID embedding, which makes it eliminated from the explicit and ensemble classes. ACARec consists of an aggregation layer, which aims to extract several contextual intents of sessions. Then, we use an attention layer to discriminatively exploit the extracted intents. Since ACARec does not have position embeddings, the attention layer is not able to distinguish the relative orders of contextual intents. Furthermore, the aggregation layer can solely leverage the items in a fixed context window, which makes ACARec is not able to learn the positional or relative orders of items in the whole session. Therefore, ACARec does not belong to any of the three categories, which allows it to explore the necessity of positional information of items. By this means, ACARec can capture the user's general intents at different time intervals within a session. To leverage the extracted intents, we use a widely-employed multi-head attention layer to recognize the main contextual intents to learn the session embedding. In short, ACARec is the first DNN-based position-agnostic approach to SBR.

To sum up, we make the following contributions:

- We propose a novel SBR model that **A**ggregates **C**ontext intents and combines with the **A**ttention mechanism, namely ACARec. To our best knowledge, ACARec is the first position-agnostic SBR model.
- We deliberately use well-known techniques to build ACARec to reveal the fact that it may be not necessary to strictly model positional information of items in SBR. We hope that ACARec inspires many session modeling studies in new ways.
- Extensive experiments demonstrate that the proposed position-agnostic model outperforms representative state-of-the-art (SOTA) implicit, explicit, and ensemble SBR models. Besides, ACARec is also efficient, achieving up to a 24.3× speedup in training efficiency.

2 Related Work

The task of SBR is first comprehensively defined by Hidasi et al. [3], so the works proposed prior to this can be considered traditional approaches.

2.1 *Traditional Models*

Due to the lack of a complete definition for the problem, early work predominantly focused on modeling the recommendation of the next item within short-term sessions. As a result, a wide variety of methods emerged.

Rule-based approaches consist of frequency rules and sequential rules. The former primarily rely on FP-Tree [2] and its variants [12, 22]. In contrast, the latter resemble the former but focus on the sequential patterns of the sub-sessions [14]. Both types require maintaining a set of rules and enumerating all rules during the inference stage. **KNN-based** [10] methods can be divided into item-KNN and session-KNN strategies, which model the similarity between items at different granularities. **MC-based** models have been widely explored [43, 21, 19], with the most influential work being FPMC [19]. **Probabilistic-based** approaches [30, 1] assume there are several hidden categories for the given items. To predict the next item, these methods first predict the hidden category and then predict the item from that category.

2.2 *Deep Neural Network Models*

DNN models follow the definition established by Hidasi et al. [3], and as a result, they consider modeling the positional information of items within sessions to be crucial for success. Based on the different strategies of modeling item positional information, DNN models can be primarily divided into three categories: implicit, explicit, and ensemble.

Implicit methods refer to position-aware models, such as RNN, which implicitly model positional information by using the previous hidden state of an item to compute its current state. In recent years, several studies [29, 17, 28] have attempted to model session data as directed graphs, where each item represents a node in the session graph, and directed edges are constructed based on the order of items. The extraction of positional information in these models relies on their structure. GRU4REC [3] employs Gated Recurrent Units (GRU) to capture the temporal information within sessions. Quadrana et al. [18] proposed utilizing hierarchical RNNs to capture multi-layer temporal information. II-RNN [20] captures both inter- and intra-session temporal patterns. SR-GNN [29] leverages gated graph neural networks to model pairwise item transitions. FGNN [17] makes use of graph attention networks (GAT) to discriminatively learn item transition patterns. In recent years, ACRec [38] attempts to use latent autocorrelation to implicitly capture temporal information of sessions.

Explicit models involve utilizing positional embeddings to represent items. These models employ addition or concatenation operations to merge item ID embeddings with their corresponding positional ID embeddings, in order to effectively represent items. STAMP [8] replaces RNN layers with attention networks. SASRec [5] also employs the attention mechanism and self-attention to learn session embeddings. Caser [24] attempts to use convolutional layers to capture temporal information. NextItNet [39] utilizes dilated convolution to learn long-term temporal dependencies of items.

Ensemble methods leverage both implicit and explicit ways to model positional information of items, i.e., they utilize both model structures (RNN, directed graphs, etc.) and positional ID embeddings. NARM [7] uses the attention mechanism to compute the importance of RNN hidden states to capture the items that affect the session intents. GC-SAN [37] combines graph convolutional layers with self-attention to utilize the advantages of both structures. S^2-DHCN [35] builds dual channels to establish self-supervised learning [9] framework to learn session representations. Moreover, some advanced techniques are introduced into session-based scenarios for recommendations [34, 26, 16, 15, 36], refer to [40] for more details.

In summary, the extraction of positional information in sessions is considered crucial for the success of DNN-based models. There is no position-agnostic DNN model has been developed so far.

3 Methodology

3.1 Problem Description

Let $I = \{i_1, i_2, \dots, i_N\}$ denote the set of items, where N represents the number of unique items. An anonymous session can be represented as $s = [i_{s,1}, i_{s,2}, \dots, i_{s,m}]$, ordered by timestamps, and $i_{s,k} \in I(1 \leq k \leq m)$ corresponds to a clicked item within session s. The objective of SBR is to predict the next item, $i_{s,m+1}$, for session s. The output of an SBR model is typically a probability distribution $y = [y_1, y_2, \dots, y_N]$ where $y_k(1 \leq k \leq N)$ indicates the predicted probability of the corresponding item i_k. The items with the top-K probabilities will be recommended.

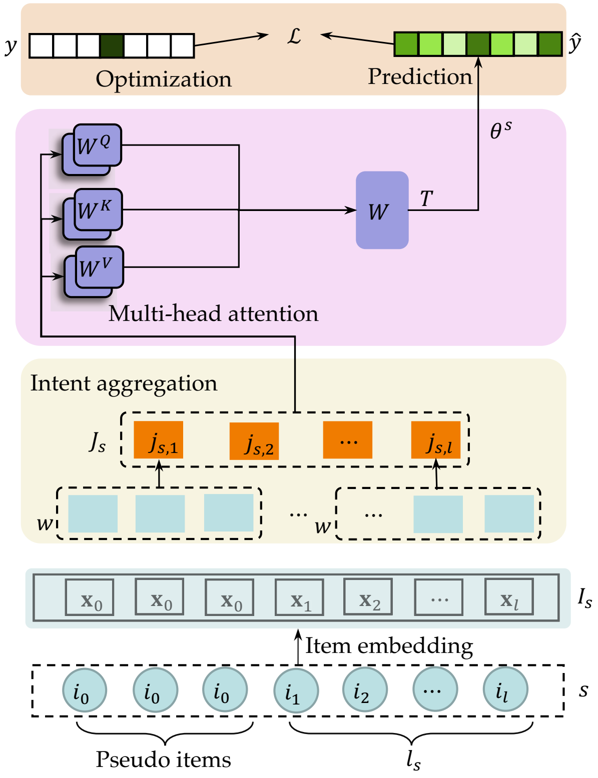


Figure 1. The architecture of ACARec, where W denotes linear transformation

3.2 Model Architecture

The architecture of ACARec is shown in Figure 1, and each component will be elaborated as follows.

3.3 Item Representation

Item Embedding. To learn the representations, we embed each item $i \in I$ into the same space. Let $\mathbf{X} \in \mathbb{R}^{N \times d}$ denote the embedding table of items, where d is the dimension of embedding space. Denote $\mathbf{x}_i \in \mathbb{R}^d$ as the i-th element in $\mathbf{X}$ and it is the vector representation of the i-th item in I. *We do not employ positional embedding on item representations, which is significantly different from the previous works.*

Input Representation. Denote the length of input session s as l_s. First, we pad each session with pseudo item i_0 to length L, where L is the maximum length of all sessions. Then, we look for the embedding table to acquire the input representation $I_s \in \mathbb{R}^{L \times d}$.

3.4 Aggregate Contextual Intent

The goal of ACARec is to capture the local contextual item preference intent, so there is no need to model positional embeddings. Consequently, one-dimensional convolution and pooling methods can effectively aggregate local session information. Moreover, compared to existing CNN-based methods [24, 41], ACARec has an important distinction in aggregating local session intent: it solely utilizes one-dimensional convolution or pooling methods to represent contextual intent, rather than using convolution combined with pooling and positional embeddings to capture time-dependent item relationships.

$$J_s = \phi_{agg}(I_s, w), \tag{1}$$

where $J_s = [j_{s,1}, j_{s,2}, \dots, j_{s,l}] \in \mathbb{R}^{l \times h}$ is the list of aggregated contextual intents, h is the hidden size, ϕ_{agg} is the aggregation function, $w > 1$ is the hyperparameter of context window size. Since there are two implementation approaches for aggregation, we will introduce them as follows.

Convolutional Aggregation. The convolutional aggregation method is denoted as ϕ_{Conv}. Given an input session I_s and window size w, for the aggregation operation ϕ^1_{Conv}, the aggregated intent can be obtained by:

$$j^1_{s,i} = \phi^1_{\text{Conv}}(I_s, w) = \sum_{k=i}^{i+w} \omega_i \odot I_{s,k}, \tag{2}$$

where ω_i represents the k^{th} learnable vector parameter in the convolution kernel, $\odot$ denotes element-wise multiplication, and $I_{s,k}$ is the representation of the k^{th} item in the input session. In practice, multiple aggregation kernels are needed. So we denote the number of kernels as h, allowing the input session to be mapped from the $\mathbb{R}^d$ space to the $\mathbb{R}^h$ hidden space.

$$j_{s,i} = \overset{h}{\underset{m=1}{||}} \phi^m_{\text{Conv}}(I_s, w), \tag{3}$$

where $||$ represents the concatenation operation for vectors. In this manner, we can aggregate the input session representation I_s into contextual intents J_s.

Pooling Aggregation. There are numerous pooling approaches but we solely select the max pooling approach, because it can efficiently aggregate the information of a context [13]. Thus, the pooling aggregation method is denoted as ϕ_{Max}.

Max pooling is similar to the convolution operation but has two key differences. Firstly, the pooling operation has no parameters. Secondly, there is no need to map it to the $\mathbb{R}^h$ space. The pooling aggregation is shown as follows.

$$\begin{aligned} \phi^i_{\text{Max}}(I_s, w) &= \max(I_{s,k}, k \in [i, \dots, i+w]), \\ j_{s,i} &= \overset{d}{\underset{m=1}{||}} \phi^m_{\text{Max}}(I_s, w). \end{aligned} \tag{4}$$

Both aggregation methods are applicable and position-agnostic.

3.5 Session Representation

We employ multi-head self-attention [25] to encode the contextual intents of the session.

Multi-Head Self-Attention (MHSA). First, we employ three different linear transformations to map the input session into hidden spaces:

$$Q = J_s W^Q + b^Q, K \quad = J_s W^K + b^K, V = J_s W^V + b^V, \tag{5}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{h \times h}$, $b^Q, b^K, b^V \in \mathbb{R}^h$ are learnable parameters. Then the self-attention is formulated as:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{h}})V. \tag{6}$$

Then we apply multi-head self-attention as follows:

$$\text{MHSA}(Q, K, V) = (\text{head}_1 || \ldots || \text{head}_\text{k}) W^O, \tag{7}$$

$$\text{where } \text{head}_\text{i}(Q, K, V) = \text{Attention}(Q_i, K_i, V_i), \tag{8}$$

where $||$ denotes concatenation operation, $W^O \in \mathbb{R}^{h \times h}$, $Q_i, K_i, V_i \in \mathbb{R}^{h \times d_k}$, $d_k = h/k$, k is the number of attention heads.

Point-wise Feed-Forward Network (PFFN). Then we apply two linear transformations with nonlinear activation function:

$$\begin{aligned} E &= \text{MHSA(Q, K, V)}, \\ T &= ReLU(EW_1 + b_1)W_2 + b_2 + E, \end{aligned} \tag{9}$$

where $W_1, W_2 \in \mathbb{R}^{h \times h}$, b_1, b_2 are bias vectors. Then we take the last representation as session interests as the previous works [3, 8] for simplicity. For clarification, even though we select the last representation, its original representation is aggregated from several items in a position-agnostic approach.

Session Representation. To this end, we can acquire context-attentive intent representation by a linear transformation:

$$\theta_s = T[l - 1]W + b, \tag{10}$$

where $\theta_s \in \mathbb{R}^d$ is the session representation, $W \in \mathbb{R}^{h \times d}$, b is the bias. We apply Dropout regularization technique to alleviate overfitting similar to [37], the dropout rate is thoroughly searched in [0,1].

Model Prediction and Training. To predict the next item, the recommendation probability can be computed by:

$$\hat{y}_i = \frac{\exp(\theta^s \cdot \mathbf{x}_i^\top)}{\sum_{j=1}^N \exp(\theta^s \cdot \mathbf{x}_j^\top)}, \tag{11}$$

where $\hat{y}$ is the output probability distribution vector, $\hat{y}_i$ is the recommendation probability of item e_i, $\mathbf{x}_i^\top$ is the transpose of item embedding vector $\mathbf{x}_i$. And the loss function $\mathcal{L}$ is formulated as:

$$\mathcal{L} = -\sum_{i=1}^N y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i), \tag{12}$$

where y is the one-hot vector of the ground truth, and the object is to minimize $\mathcal{L}$.

4 Experiments

4.1 Experimental Settings

Dataset and Implementation Detail. We evaluate ACARec on two real-world benchmark datasets, *RetailRocket*[1] and *Tmall*[2], which have been extensively used in the previous studies. RetailRocket is a dataset on a Kaggle contest published by an e-commerce company. Tmall dataset comes from IJCAI-15 competition, which contains anonymized shopping logs on Tmall online shopping platform. The same as the previous works [7, 29, 37, 35], we first remove the sessions that contain only one item and the items that occurred less than five times. Then, we augment the datasets by sequence splitting. Specifically, given a session $s = [i_{s,1}, i_{s,2}, \ldots, i_{s,m}]$, we split it by sequence to generate sub-session and the corresponding labels $([i_{s,1}], i_{s,2}), ([i_{s,1}, i_{s,2}], i_{s,2}), \ldots, ([i_{s,1}, \ldots, i_{s,m-1}], i_{s,m})$. Some statistics of the preprocessed two datasets RetailRocket and Tmall are shown in Table 2.

The implementation details are as follows: we set the batch size as 100, item embedding size d as 100, and the optimizer is Adam [6] with a learning rate of 0.001. The hidden size h is set as 256. All experiments in this paper are conducted on an NVIDIA GTX 1080 Ti. We execute the model five times and report the average performance.

Baseline and Evaluation Protocols. Since we aim to investigate the necessity of modeling positional information of items rather than propose a SOTA model, we select the following representative and competitive traditional, implicit, explicit, and ensemble SBR models as our baselines:

- FPMC [19] is a traditional SBR model based on Markov Chain. The original version has user profile modeling components, we do not consider user information.
- GRU4REC [3] is the first SBR model that is based on RNN, and it is an implicit model.
- NARM [7] is an ensemble approach that first combines the attention mechanism and RNN to capture the main purpose of sessions.
- STAMP [8] explicitly utilizes the self-attention mechanism to replace RNN to model the long-term interest of sessions.
- SASRec [5] is also an explicit method that is solely based on the self-attention mechanism. It is a contemporaneous work with STAMP.
- NextItNet [41] is an explicit model and it is the best-performing CNN-based SBR model. This model employs dilated convolution layers, which enable it to effectively learn and capture long-term dependencies among items within a given session.
- SR-GNN [29] is an implicit model as well as the first SBR approach based on GNN.
- GC-SAN [37] combines GNN and multi-layer self-attention to make recommendations.
- GCE-GNN [28] is an ensemble method that learns global and local information of sessions in different views. It can leverage intra-session patterns.
- S^2-DHCN [35] is an ensemble method and it combines hypergraph and self-supervised learning to enhance session-based recommendation and alleviate the data sparsity issue.

We reproduce all the baseline models on the same device as ACARec. For each metric, we report the best result of the original paper and our reproduced result. Following the prior works [8, 29, 35], we use P@K (Precision) and MRR@K (Mean Reciprocal Rank) as the evaluation metrics of recommendation results where K is 10 or 20.

$$P@K = \frac{1}{|S|} \sum_{s \in S} \frac{\sum_{i=1}^K \mathbb{1}(s(i))}{K}, \tag{13}$$

where S denotes the set of all sessions, and $\mathbb{1}$ is the indicator function that yields 1 when the item in the function is the ground truth item

[1] https://www.kaggle.com/retailrocket/ecommerce-dataset

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=42

Table 1. Performance comparison on two datasets (%). In each metric, the best result is highlighted in boldface and the second best is underlined. And † indicates statistic significant improvement over all baseline models for t-test with p-value < 0.01.

Method	RetailRocket				Tmall			
	P@10	MRR@10	P@20	MRR@20	P@10	MRR@10	P@20	MRR@20
FPMC	25.99	13.38	32.37	13.82	13.10	7.12	16.06	7.32
GRU4REC	34.41	15.06	44.89	15.77	14.16	6.56	18.20	6.85
NARM	42.07	24.88	50.22	24.59	19.17	10.42	23.30	10.70
STAMP	42.95	24.61	50.96	25.17	22.63	13.12	26.47	13.36
SASRec	44.65	25.53	51.12	25.91	22.06	14.02	26.95	14.21
NextItNet	41.12	23.99	48.26	24.48	22.67	13.12	27.22	13.32
SR-GNN	43.21	26.07	50.32	26.57	23.41	13.45	27.57	13.72
GC-SAN	44.54	27.18	51.63	27.72	21.32	12.43	25.38	12.72
GCE-GNN	46.05	27.48	53.63	28.01	28.02	15.08	33.42	15.42
S^2-DHCN	46.15	26.85	53.66	27.30	26.22	14.60	31.42	15.05
ACARec	**47.76**†	**29.09**†	**55.39**†	**29.71**†	**28.99**†	**17.41**†	**33.75**†	**17.68**†

Table 2. Statistics of RetailRocket and Tmall Datasets

Dataset	# training	# test	# items	Avg. Len.
RetailRocket	433,643	15,132	36,968	5.43
Tmall	351,268	25,898	40,728	6.69

and 0 otherwise. $s(i)$ is the session item ranked i in the score of the item recommended for session s.

$$MRR@K = \frac{1}{|S|}\sum_{s\in S}\frac{1}{\text{rank}(s_{tgt})}, \tag{14}$$

where s_{tgt} denotes the target value of the next item in session s, corresponding to the ground truth item label for the next item in the training session. Additionally, rank$(\cdot)$ represents the ranking of the target item within the recommended list.

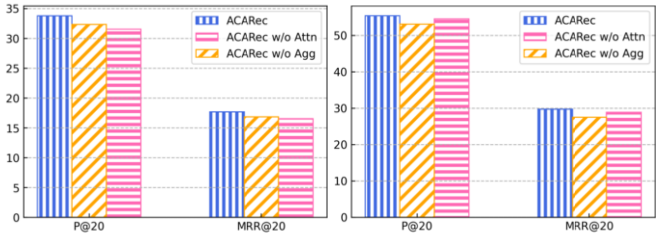
Research Questions. In this paper, we would like to answer the following Research Questions (RQs):

- RQ1: Can the first position-agnostic DNN model beat representative traditional, implicit, explicit, and ensemble models?
- RQ2: Is every constructed module in ACARec model useful?
- RQ3: Have we brought some insights on modeling session preference for SBR?
- RQ4: Is ACARec efficient for the task of SBR?

4.2 Overall Results (RQ1)

The overall results are shown in Table 1, we can observe that ACARec consistently outperforms the baseline models and achieves statistically significant improvement in each metric. We can draw the following conclusions based on the results.

- GRU4REC outperforms the traditional FPMC model on the majority of evaluation metrics. It corroborates the view that DNN-based SBR methods can alleviate the data sparsity issue, which is difficult for traditional methods to overcome.
- SASRec is the best-performing attention-based method, particularly in the RetailRocket dataset, where it demonstrates a significant improvement in P@10 compared to STAMP. This suggests that even within similar approaches, differences in implementation techniques can lead to substantial variations in results.



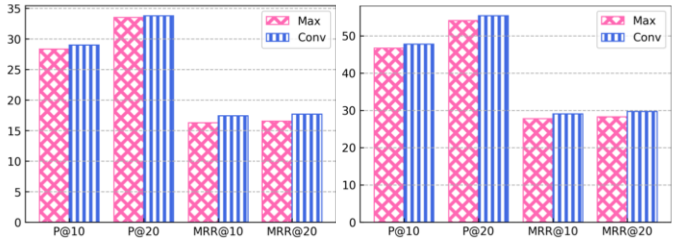
(a) Tmall (b) RetailRocket

Figure 2. Ablation study

- NextItNet consistently outperforms GRU4REC, but it scores lower on most metrics compared to attention-based SBR methods. This indicates that capturing long-term dependencies within sessions for SBR problems is challenging. However, for the Tmall dataset, NextItNet slightly surpasses all attention-based models in terms of P@10 and P@20 metrics. This suggests that even if long-term dependencies cannot be captured within sessions, relaxing the assumption that models require strict position information for items might help capture session intent.
- GNN-based methods (SR-GNN, GC-SAN, GCE-GNN, and S^2-DHCN) outperform traditional (FPMC), RNN-based (GRU4REC), attention-based (NARM, STAMP, and SASRec), and CNN-based (NextItNet) methods in most metrics. It further indicates that relaxing the strictly ordered sequence modeling strategy is beneficial to SBR.
- The proposed model, ACARec, consistently outperforms all traditional, implicit, explicit, and ensemble baseline models. This achievement can be deemed as the joint efforts of contextual intents aggregation and discriminatively utilization of these intents in a position-agnostic way.

4.3 Ablation Study (RQ2)

To investigate the contribution of each component in ACARec, we develop the following variants of ACARec: **ACARec w/o Agg** and



(a) Tmall (b) RetailRocket

Figure 3. Aggregation approaches comparison

ACARec w/o Attn. The former removes the context intent aggregation and the latter removes the attentive intent encoding layer. The performance of the two variants on Tmall is depicted in Figure 2. Based on the observation, we can draw the following conclusions:

- Compared to ACARec, both variant models exhibit reduced performance, but they still outperform most of the baselines. This indicates that these two components provide benefits in capturing session intent within ACARec.
- On Tmall, the aggregation operation can achieve better session intent capture compared to using attention mechanism modules. However, the opposite results are observed in the RetailRocket dataset. A possible reason for this discrepancy is the influence of data distribution, as both datasets come from the real world, with Tmall's data primarily originating from China and RetailRocket serving multiple countries. However, combining both modules can lead to performance improvement compared to using a single module alone.
- It demonstrates that these two components are beneficial to SBR, and aggregating operations can truly capture better session intents than sequential and graph-based models, which indicates our suggestion to some extent.

4.4 Model Analysis (RQ3)

Aggregation Approaches. Even though our approach is position-agnostic from the perspective of the whole session, convolutional operations have positional settings in aggregating the local contextual intents of the fixed window of items. To eliminate this trivial argument, we use the entirely position-agnostic pooling method to validate our arguments. We denote the model that employs the *1-D convolutional neural networks* and the *max pooling* operation, where the former is denoted as *Conv* and the latter as *Max*, respectively. The performance comparison of these two models is shown in Figure 3, and we can yield the following conclusions:

- On both datasets, the performance of *Conv* consistently surpasses *Max* across all metrics. This could be attributed to the fact that the convolutional operation has more parameters than the pooling method, and *Conv* possesses greater representation capability for capturing contextual information. As a result, ACARec employs *Conv* by default.
- *Even employing the simple max pooling method as the aggregation function, it is still capable of outperforming most baseline models.* This strongly suggests that not modeling the positional information of items is a viable approach for SBR.
- In summary, the max pooling aggregation surpasses most of the competing models, indicating that strictly modeling the positional embedding for items may harm capturing session intent. The fact that one-dimensional convolution outperforms pooling suggests that modeling local context item preferences provides a better inductive bias than not modeling positional embeddings at all.

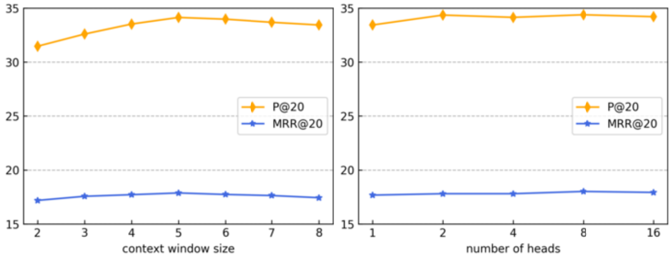
Hyperparameter Study. In the ACARec model, there are two important hyperparameters: one is the context intent aggregation window size w, and the other is the number of attention heads k used in the attention mechanism utilizing the context intent. The settings of these two hyperparameters may impact the model's robustness. Therefore, we separately sample these parameters and present their effects under different configurations.

For the number of attention heads k, we sample values from $\{1, 2, 4, 8, 16\}$. For the context window size w, values are sampled from the range $\{2, 3, 4, 5, 6, 7, 8\}$. Experiments are then conducted on the RetailRocket and Tmall datasets. The hyperparameter exploration results on Tmall are shown in Figure 4, and the effects on RetailRocket are depicted in Figure 5. A simple observation reveals different impact trends for the two hyperparameters on the model's performance. We analyze the experimental results as follows:

- On Tmall, as the number of attention heads k increases, the performance of ACARec displays a trend of initially rising, then declining after reaching its peak, and eventually deteriorating again. With the increase of the context window size w, the ACARec performance exhibits a pattern of rising to a peak value before decreasing.
- On RetailRocket, as both the number of attention heads k and the context window size w increase, the performance of ACARec exhibits a pattern of initially rising to a peak value followed by a decline.
- These experimental results suggest that ACARec is not highly sensitive to the number of attention heads. Additionally, when the context window size is greater than 5, ACARec is not very sensitive to the increase in context window size, as evidenced by the experimental outcomes. A possible reason is that when the context window is relatively small, the contextual intent contains less information.
- The best-performing hyperparameters on the two datasets are not identical, but they are very close: the optimal number of attention heads is 8 for both the Tmall and RetailRocket datasets. The best window size in Tmall is 5, while in RetailRocket, the optimal window size w is 4, but the performance decline at $w = 5$ is minimal. This demonstrates that the proposed model in this chapter is not sensitive to variations in hyperparameters and exhibits well robustness.
- Although altering the values of these two hyperparameters may impact ACARec's performance, resulting in a slight decline in the model's effectiveness, the reduced performance remains competitive. The fluctuating outcomes are still superior to those of most baseline models.

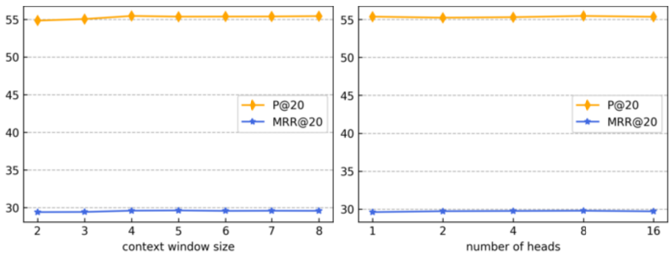
4.5 Efficiency Comparison (RQ4)

To evaluate the efficiency of ACARec, we compare the training time per epoch and trainable parameters with recent SOTA models on the same device. The results are shown in Table 3. Based on the results, we can obtatin the following observation:



(a) Context window size (b) The number of attention heads

Figure 4. Hyperparameter study on Tmall.



(a) Context window size (b) The number of attention heads

Figure 5. Hyperparameter study on RetailRocket.

- Compared to the baseline methods, the ACARec model proposed in this chapter achieves 2.7× - 9.6× and 3.1× - 24.3× acceleration in training speed on the RetailRocket and Tmall datasets, respectively. This highlights the advantage of ACARec in terms of training efficiency.
- While ACARec is more effective than the SOTA methods, it involves more training parameters. However, it has at most about 20% more parameters compared to the baseline models. These additional parameters are introduced by the local context aggregation component, and the number of extra parameters is not significant when compared to traditional approaches.
- It can be observed that ACARec w/o Agg has more parameters and slower execution speed than ACARec w/o Attn. Additionally, ACARec w/o Attn is the fastest method in terms of per-epoch training speed and the most lightweight approach in terms of the number of parameters.
- Considering the experimental results in Figure 2, even using only the last aggregated intent can outperform the baselines in most metrics. Thus, it can be concluded that aggregating context intent is efficient in terms of both performance and effectiveness. In addition, exploring more effective structures for utilizing context intent could be a potential future work.

5 Conclusion

Table 3. Training time per epoch and the number of trainable parameters, where s, m, and M respectively represent second, minute, and million. (Due to the memory issue, GCE-GNN's batch size on RetailRocket is set as 50)

Method	RetailRocket		Tmall	
	Time	#Params	Time	#Params
NextItNet	44m3s	3.85M	29m12s	4.23M
SR-GNN	24m41s	3.86M	5m17s	4.23M
GC-SAN	12m20s	3.87M	3m42s	4.24M
GCE-GNN	38m40s	3.98M	2m32s	4.35M
S^2-DHCN	2h26m	3.94M	1h2m	4.31M
ACARec	**4m35s**	4.62M	**1m12s**	5.02M
ACARec w/o Agg	4m17s	4.54M	1m8s	4.92M
ACARec w/o Attn	1m10s	3.83M	41s	4.20M

In this paper, we propose a novel and concise SBR model called ACARec to investigate the necessity of positional information modeling in SBR. We extract contextual intents instead of modeling strictly positional information of items. Then, we learn session representation by discriminatively leveraging the extracted contextual intents. Extensive experiments demonstrate ACARec's superiority in terms of effectiveness and efficiency. As a result, our proposed model reveals that the intent of sessions may not be affected by the strict orders of items. We hope our work inspires a variety of session intent representation methods in novel ways. In future work, we plan to explore lightweight structures for utilizing contextual intents.

6 Acknowledgement

This research is supported by National Natural Science Foundation of China (Grant No.62276154), Research Center for Computer Network (Shenzhen) Ministry of Education, Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No. 2023A1515012914), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JSGG20210802154402007), the Major Key Project of PCL for Experiments and Applications (PCL2021A06), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (HW2021008).

References

[1] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, and Yeow Meng Chee, 'Personalized ranking metric embedding for next new poi recommendation', in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 2069–2075. ACM, (2015).

[2] Jiawei Han, Jian Pei, and Yiwen Yin, 'Mining frequent patterns without candidate generation', *ACM sigmod record*, **29**(2), 1–12, (2000).

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, 'Session-based recommendations with recurrent neural networks', in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, (2016).

[4] Dietmar Jannach and Malte Ludewig, 'When recurrent neural networks meet the neighborhood for session-based recommendation', in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 306–310, (2017).

[5] Wang-Cheng Kang and Julian McAuley, 'Self-attentive sequential recommendation', in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206. IEEE, (2018).

[6] Diederik P. Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, (2015).

[7] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma, 'Neural attentive session-based recommendation', in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 1419–1428, (2017).

[8] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang, 'Stamp: short-term attention/memory priority model for session-based recommendation', in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1831–1839, (2018).

[9] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang, 'Self-supervised learning: Generative or contrastive', *IEEE Transactions on Knowledge and Data Engineering*, (2021).

[10] Malte Ludewig and Dietmar Jannach, 'Evaluation of session-based recommendation algorithms', *User Modeling and User-Adapted Interaction*, **28**(4-5), 331–390, (2018).

[11] Tom M Mitchell, *The need for biases in learning generalizations*, Citeseer, 1980.

[12] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa, 'Effective personalization based on association rule discovery from web usage data', in *Proceedings of the 3rd international workshop on Web information and data management*, pp. 9–15, (2001).

[13] Lili Mou, Ge Li, Lu Zhang, Tao Wang, and Zhi Jin, 'Convolutional neural networks over tree structures for programming language processing', in *Thirtieth AAAI conference on artificial intelligence*, (2016).

[14] Utpala Niranjan, RBV Subramanyam, and V Khanaa, 'Developing a web recommendation system based on closed sequential patterns', in *Information and Communication Technologies: International Conference, ICT 2010, Kochi, Kerala, India, September 7-9, 2010. Proceedings*, pp. 171–179. Springer, (2010).

[15] Kai Ouyang, Xianghong Xu, Miaoxin Chen, Zuotong Xie, Hai-Tao Zheng, Shuangyong Song, and Yu Zhao, 'Mining interest trends and adaptively assigning sample weight for session-based recommendation', in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, p. 2174–2178, New York, NY, USA, (2023). Association for Computing Machinery.

[16] Kai Ouyang, Xianghong Xu, Chen Tang, Wang Chen, and Haitao Zheng, 'Social-aware sparse attention network for session-based social recommendation', in *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 2173–2183, (2022).

[17] Ruihong Qiu, Jingjing Li, Zi Huang, and Hongzhi Yin, 'Rethinking the item order in session-based recommendation with graph neural networks', in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 579–588, (2019).

[18] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi, 'Personalizing session-based recommendations with hierarchical recurrent neural networks', in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pp. 130–137, (2017).

[19] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme, 'Factorizing personalized markov chains for next-basket recommendation', in *Proceedings of the 19th international conference on World wide web*, pp. 811–820, (2010).

[20] Massimiliano Ruocco, Ole Steinar Lillestøl Skrede, and Helge Langseth, 'Inter-session modeling for session-based recommendation', in *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pp. 24–31, (2017).

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

[22] Bo Shao, Dingding Wang, Tao Li, and Mitsunori Ogihara, 'Music recommendation based on acoustic features and user access patterns', *IEEE Transactions on Audio, Speech, and Language Processing*, **17**(8), 1602–1611, (2009).

[23] Yong Kiam Tan, Xinxing Xu, and Yong Liu, 'Improved recurrent neural networks for session-based recommendations', in *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 17–22, (2016).

[24] Jiaxi Tang and Ke Wang, 'Personalized top-n sequential recommendation via convolutional sequence embedding', in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 565–573, (2018).

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, 'Attention is all you need', in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, (2017).

[26] Liuyin Wang, Xianghong Xu, Kai Ouyang, Huanzhong Duan, Yanxiong Lu, and Hai-Tao Zheng, 'Self-supervised dual-channel attentive network for session-based social recommendation', in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2034–2045. IEEE, (2022).

[27] Shoujin Wang, Longbing Cao, Yan Wang, Quan Z Sheng, Mehmet A Orgun, and Defu Lian, 'A survey on session-based recommender systems', *ACM Computing Surveys (CSUR)*, **54**(7), 1–38, (2021).

[28] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu, 'Global context enhanced graph neural networks for session-based recommendation', in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 169–178, (2020).

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

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

[31] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip, 'A comprehensive survey on graph neural networks', *IEEE transactions on neural networks and learning systems*, **32**(1), 4–24, (2020).

[32] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip, 'A comprehensive survey on graph neural networks', *IEEE transactions on neural networks and learning systems*, **32**(1), 4–24, (2020).

[33] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui, 'Self-supervised graph co-training for session-based recommendation', in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2180–2190, (2021).

[34] Xin Xia, Hongzhi Yin, Junliang Yu, Yingxia Shao, and Lizhen Cui, 'Self-supervised graph co-training for session-based recommendation', in *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 2180–2190, (2021).

[35] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang, 'Self-supervised hypergraph convolutional networks for session-based recommendation', in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*, pp. 4503–4511, (2021).

[36] Xin Xia, Junliang Yu, Qinyong Wang, Chaoqun Yang, Nguyen Quoc Viet Hung, and Hongzhi Yin, 'Efficient on-device session-based recommendation', *ACM Transactions on Information Systems*, **41**(4), 1–24, (2023).

[37] Chengfeng Xu, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Fuzhen Zhuang, Junhua Fang, and Xiaofang Zhou, 'Graph contextualized self-attention network for session-based recommendation.', in *IJCAI*, pp. 3940–3946, (2019).

[38] Xianghong Xu, Kai Ouyang, Liuyin Wang, Jiaxin Zou, Yanxiong Lu, Hai-Tao Zheng, and Hong-Gee Kim, 'Modeling latent autocorrelation for session-based recommendation', in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 4605–4609, (2022).

[39] Ghim-Eng Yap, Xiao-Li Li, and Philip S Yu, 'Effective next-items recommendation via personalized sequential pattern mining', in *Database Systems for Advanced Applications: 17th International Conference, DASFAA 2012, Busan, South Korea, April 15-19, 2012, Proceedings, Part II 17*, pp. 48–64. Springer, (2012).

[40] Junliang Yu, Hongzhi Yin, Xin Xia, Tong Chen, Jundong Li, and Zi Huang, 'Self-supervised learning for recommender systems: A survey', *IEEE Transactions on Knowledge and Data Engineering*, (2023).

[41] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M Jose, and Xiangnan He, 'A simple convolutional generative network for next item recommendation', in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 582–590, (2019).

[42] Elena Zheleva, John Guiver, Eduarda Mendes Rodrigues, and Nataša Milić-Frayling, 'Statistical models of music-listening sessions in social media', in *Proceedings of the 19th international conference on World wide web*, pp. 1019–1028, (2010).

[43] Andrew Zimdars, David Maxwell Chickering, and Christopher Meek, 'Using temporal data for making recommendations', *arXiv preprint arXiv:1301.2320*, (2013).