CoANBR: A Collaborative Aggregation Model for Next Basket Recommendation with Time-independent Sequence Modeling

Li Lin[1], Kaiwen Xia[1,2](✉), Haotian Shen[1], and Shuai Wang[1](✉)

[1] Southeast University, Nanjing, China
[2] JD logistics, Beijing, China
`{linli321,kevinxia,213221090,shuaiwang}@seu.edu.cn`

Abstract. The Next Basket Recommendation (NBR) task focuses on predicting a set of items for users based on their interests. Although utilizing preference information from users' historical sequences can enhance NBR accuracy by capturing general preferences, these methods struggle when historical data is sparse. Additionally, modeling customized preferences for different users remains a challenge. In this work, we introduce a novel **Co**llaborative **A**ggregation Model for NBR with Time-independent Sequence Modeling, namely **CoANBR**, which comprises three modules: (1) The Item-aware Aggregator constructs an item-item graph using user-item and set-item interactions, leveraging graph learning to obtain item embeddings and effectively aggregate neighborhood information. (2) The User Preference Enhancer captures interactions among items to help users discover potential preferred items globally, and further enhances general preferences to address data sparsity issues. (3) The Decoupled Temporal Attention Module models the temporal dependencies of item positional information within interaction sequences, enabling the customization of user preferences across different contexts. Extensive experiments on real-world datasets show CoANBR's effectiveness, consistently outperforming several state-of-the-art NBR methods.

Keywords: Next Basket Recommendation · Collaborative Aggregation · User Modeling.

1 Introduction

Recommendation systems play a vital role in online shopping platforms [1]. In the shopping landscape, users often purchase multiple items simultaneously due to diverse demands or promotional events [13], making Next Basket Recommendation (NBR) [22] particularly important on online shopping platforms like Amazon, Taobao, and JD. Unlike single-item recommendation [4,32] or sequential recommendation [38,36], NBR aims to recommend a set of items based on the user's historical set sequences, without requiring a strict order.

✉ Kaiwen Xia and Shuai Wang are corresponding authors.

In recent years, NBR has been widely applied in recommendation systems. Early efforts primarily focused on modeling the context of item sets [22,33], while more recent work emphasizes user preferences and their interactions with these item sets [5,34]. Although recent researchers have improved NBR accuracy by incorporating preference from user history sequences [16,15], these methods may be limited when users lack sufficient historical interactions (i.e., sparse user data). In this work, we emphasize the importance of incorporating user-item set collaboration in NBR, including enhancing general user preferences through item sets and modeling their customized preferences for items.

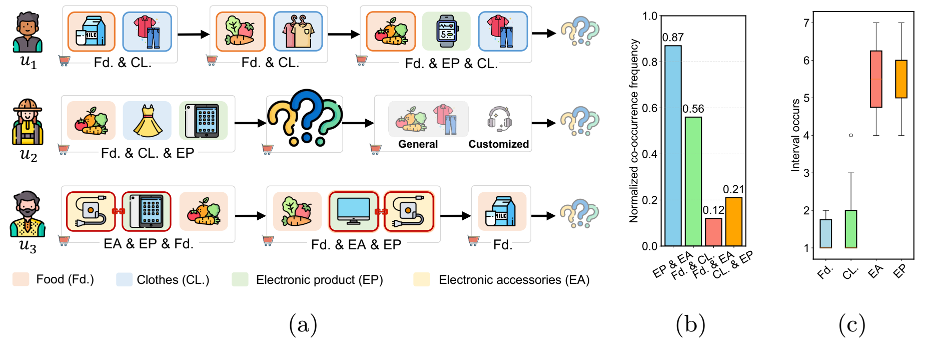


Fig. 1: A toy example of user-item set collaboration is illustrated in the next basket recommendation. (a) Sequences for users u_1, u_2, and u_3, with u_2 being sparse. Item combinations for u_1 and u_3 suggest general (Fd. and CL.) and customized (EA) preferences. (b) The normalized co-occurrence frequency between items. (c) The intervals at which items appear within the set sequences of the same user, indicating the user's demand pattern for the items.

Figure 1 illustrates how user-item set collaboration influences the learning of user preferences in NBR, particularly when there is significant variation in the sparsity of historical sequences among users. Firstly, interactions between users and item sets, along with contextual information, offer insights into enhancing general user preferences. When electronic products and accessories frequently appear together for user u_3, it indicates a strong correlation (as shown in Figure 1b). Furthermore, the sequences of sets for u_1 and u_3 show that once food and clothes are present, they are likely to appear in subsequent sets as well (as shown in Figure 1c). Therefore, in the limited set sequence of u_2, if food and clothes appear, they are also likely to appear in the next set of u_2, demonstrating a classic collaborative filtering phenomenon [30]. Secondly, the item set sequence can reflect users' customized preferences, allowing for the recommendation of appropriate sets at the right time. For example, user u_1 has cyclical demands for clothing but random demands for electronics. Thus, by observing these patterns, we can enhance user preference features and customize recommendations to improve practicality.

Despite the advantages of user-item set collaboration in NBR, its application faces two challenges. Firstly, *capturing and enhancing the general preference fea-*

tures of users with insufficient behavioral data is non-trivial. Existing NBR recommendation methods [2,5,34] mainly rely on the intra-sequence information to encode set sequences for user features. However, when user sequences are sparse, ensuring the effectiveness of user feature representation becomes difficult [7]. This issue is further exacerbated when item data exhibits a long-tail distribution [37,3]. Secondly, *considering the order of items within different sets and learning the relative positional relationships between sets and items are crucial for further customizing user preferences.* Unlike sequential recommendations, converting item timestamps into context vectors fails to adequately emphasize set context information, as time intervals influence the relative position of user-item interactions [14,25,23]. For example, the effects of a one-day gap versus a one-month gap can be vastly different, which makes it unreasonable to embed only the relative value of the number of interaction set sequences. Also, introducing relative positions enhances the linkage of items in the same set since the items in the same set interact at the same timestamp.

To address the above challenges, we propose a **C**ollaborative **A**ggregation Model for **N**ext **B**asket **R**ecommendation with time-independent sequence modeling, named **CoANBR**. To leverage user-item set collaboration for enhancing user preference features, we first develop an *Item-aware Aggregator*, which constructs item-item graphs by similarity and performs message aggregation on the graphs to fuse the item's neighbor information. Subsequently, a *User Preference Enhancer* uses a self-attention mechanism to capture interaction information between items, allowing users to query potentially interesting items on a global scale and significantly mitigate the effects of sparsity. To tailor customized user preferences, we design a *Decoupled Temporal Attention Module* that integrates absolute position and relative time to capture the time-independent features of items. The contributions of our work can be summarized as follows:

– To the best of our knowledge, this is the first next-basket recommendation model that enhances sparse user sequences through user-item set collaboration to learn user preferences.
– The proposed CoANBR incrementally learns user preferences. Initially, it aggregates and enhances item information to capture general user preferences. Then, it integrates absolute position, relative time, and item information to obtain customized user preferences.
– Extensive experiments are conducted on three benchmark datasets and the results demonstrate the effectiveness of our method both on dense and sparse datasets compared with the state-of-the-art.

2 Preliminaries and Problem Formulation

Definition 1: User-Item Matrix. The user-item matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ represents interactions between m users and n items, with $a_{i,j}$ denoting the number of interactions between user u_i and item v_j.
Definition 2: Item-Item Graph. The item-item graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is constructed by connecting items based on their co-occurrence in the user-item ma-

trix **A**. Here, $\mathcal{V}$ denotes the item set, and $\mathcal{E}$ denotes the edge set. For each edge $e_{j,j'} \in \mathcal{E}$, the weight between any two items v_j, $v_{j'}$ is defined by Eq.2 and Eq.3.
Problom: Next Basket Recommendation. Let $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ and $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ denote the sets of m users and n items, respectively. Each user $u_i \in \mathcal{U}$ has an interaction sequence represented as $\mathcal{B}_u(t) = \{\mathbf{b}_u^{t_1}, \mathbf{b}_u^{t_2}, \ldots, \mathbf{b}_u^{t_n}\}$, where $\{t_1, t_2, \ldots, t_n\}$ are the corresponding timestamps, and $\mathbf{b}_u^{t_n}$ refers to the set of items interacted with at timestamp t_n. Furthermore, the next basket recommendation problem can be described as follows: given the historical interaction sequence $\mathcal{B}_u(t)$ of user u_i, recommend item set $\mathbf{b}_u^{t_{n+1}}$ they are likely to interact with at timestamp t_{n+1}.

3 Methodology

3.1 Architecture Overview

Figure 1 shows the architecture of our proposed CoANSR, consisting of the following components: the Item-aware Aggregator, the User Preference Enhancer, the Decoupled Temporal Attention Module, and the Output Layer. The process begins with constructing item-item graphs based on the co-occurrence information of items within the interaction sequence. The Item-aware Aggregator then aggregates messages on the graph to incorporate neighborhood information for each item. Subsequently, the User Preference Enhancer captures interactions between items, allowing users to query potentially interesting items on a global scale and effectively alleviating the impact of data sparsity. Following this, the Decoupled Temporal Attention Module models the temporal correlations of positional information within the interaction sequence, including both absolute and relative time positions, to customize user preferences and recommend the next item set.

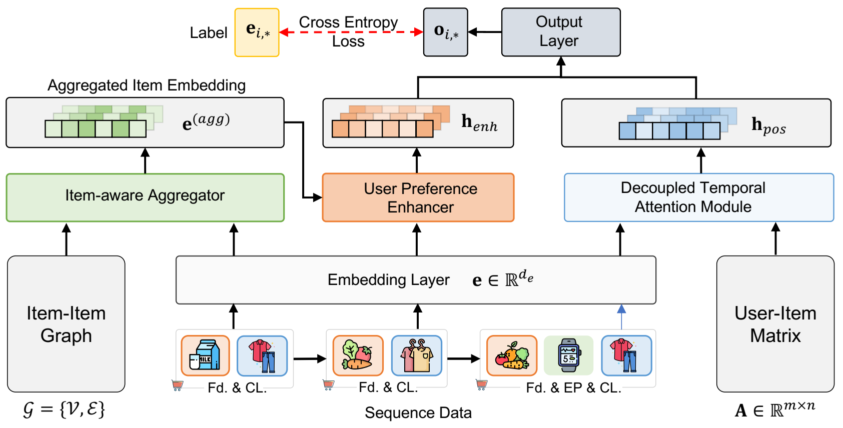


Fig. 2: The architecture of our proposed CoANSR.

3.2 Item-aware Aggregator

In the next basket recommendation, we identify pairs of items with high co-occurrence frequency within item sets, which is applicable even when users' historical set sequences are sparse. This is because user collaboration can also be expressed through item similarity. Therefore, in the user-item set collaborative process, it is crucial to aggregate the relationships between items both outside and within sets. As shown in Figure 3, Item-aware Aggregator includes *Item-Item Graph Construction* and *Item Neighboring Aggregation.*

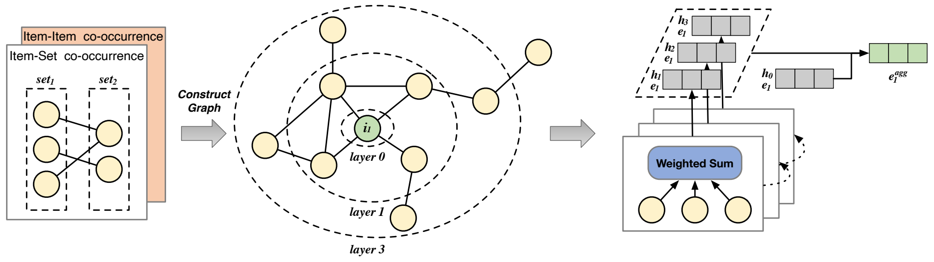


Fig. 3: The detailed architecture of item-aware aggregator. It consists of two steps: (1) the construction of the item-item graph construction and (2) the item neighboring aggregation.

Item-item Graph Construction. In this work, we calculate item similarity by constructing an item-item matrix, which has been validated in related modeling [18,8]. Unlike previous work, we use the user-item interaction matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ to construct an item-item similarity matrix $\mathbf{A}^O$ outside of sets. The corresponding item-item graph is denoted as $\mathcal{G}^O = \{\mathcal{V}, \mathbf{A}^O\}$, which preserves multiple interactions by users with the same item, reflecting their preferences. The item-item similarity matrix outside of sets, $\mathbf{A}^O \in \mathbb{R}^{n \times n}$, is obtained through the following equation:

$$\mathbf{A}^O = \mathbf{A}^T \mathbf{A} \tag{1}$$

To capture higher-order connectivity between items, inspired by [19], we optimize the values in the matrix $\mathbf{A}^O$ to approximate the graph's convergence state. The similarity between v_j and $v_{j'}$ is adjusted as follows:

$$A^O_{j,j'} \leftarrow \frac{A^O_{j,j'}}{d_i - A^O_{j,j'}} \sqrt{\frac{\xi_j}{\xi_{j'}}} \tag{2}$$

where ξ_j and $\xi_{j'}$ denote the degrees of v_j and $v_{j'}$ in $\mathbf{A}^O$ (summed by column). However, only calculating item similarity solely based on user collaboration information overlooks the influence of items within sets. It is counterintuitive to assume that the similarity between two items should be indifferent to their presence in a set. This assumption holds as long as they appear simultaneously in a user's interaction sequence.

To enhance the relevance of items within sets, we further analyze the collaborative information within sets to compute similarities between items. Specifically, by utilizing the relation $v_j \in \mathbf{b}_u^t$, we construct the set-item matrix $\mathbf{A}^S$ and obtain the item-item similarity matrix $\mathbf{A}^W \in \mathbb{R}^{n \times n}$ within sets through its transpose. Ultimately, we derive the final item-item similarity matrix by $\mathbf{A}^O$ and $\mathbf{A}^W$:

$$\hat{\mathbf{A}} = \alpha \mathbf{A}^O + (1-\alpha)\mathbf{A}^W = \alpha \mathbf{A}^O + (1-\alpha)\mathbf{A}^S {\mathbf{A}^S}^\top, \quad \alpha \in [0,1] \tag{3}$$

Subsequently, we construct the final item-item graph $\hat{\mathcal{G}} = \{\mathcal{V}, \hat{\mathcal{E}}\}$ based on $\hat{\mathbf{A}}$. The constructed item-item graph lacks sparsity, which increases the computational cost of graph convolutions and introduces unnecessary noise, complicating model optimization and training. To ensure the sparsity of the item-item graph and improve training efficiency, we retain only the top-k similar items for each item, allowing subsequent graph convolutions to integrate more significant neighbors.

Item Neighboring Aggregation. After constructing the item-item graph, we utilize a graph convolution module to aggregate the neighbors of items in order to enhance their embedding representations. For each item i, its initial embedding $\mathbf{e}_i \in \mathbb{R}^{d_e}$ is concatenated with encoded item ID and category ID, with d_e representing the length of the embedding. Since the item-item graph only contains the ID embeddings of the items and lacks other attributes, we omit the use of feature transformation and nonlinear activation. Instead, we directly perform linear weighted aggregation while considering the edge weight information between each pair of items in $\hat{\mathcal{G}}$. The propagation rule for v_i is as follows:

$$\mathbf{e}_i^{(l+1)} = \sum_{k \in \mathcal{N}_i} \frac{\hat{A}_{i,k}}{\sqrt{|\mathcal{N}_i|}\sqrt{|\mathcal{N}_k|}} \mathbf{e}_k^{(l)} \tag{4}$$

where $|\mathcal{N}_i|$ and $|\mathcal{N}_k|$ denote the number of neighbors for v_i and v_k in the item-item graph. Furthermore, we conduct L layers of message passing, and then each layer's obtained embeddings are weighted and averaged to produce the final embedding. The final item embedding output is:

$$\mathbf{e}_i^{(agg)} = \frac{1}{L+1} \sum_{l=0}^{L} \mathbf{e}_i^{(l)} \tag{5}$$

3.3 User Preference Enhancer

Similar to the initial item embedding, for a user i, the initial embedding $\mathbf{h} \in \mathbb{R}^{d_u}$ is derived from the user ID, containing only the encoded information needed to distinguish users. In contrast, the item embeddings output by the Item-aware Aggregator include the user's general preferences in interacting with both sets and items. To transfer the beneficial information from item embeddings to enhance the user's initial embedding, the User Preference Enhancer employs an enhanced attention mechanism, integrating this information into the user embedding. Specifically, we compute the synergy between all items and any user to enhance the user's general preference for items. This preference, represented by the enhanced user feature, is calculated as follows:

$$\mathbf{h}_{enh} = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{W}_Q(\mathbf{K}\mathbf{W}_K)^\top}{\sqrt{m}}\right)(\mathbf{e}^{(agg)}\mathbf{W}_V) \tag{6}$$

where m denotes the number of users, $\mathbf{Q}, \mathbf{K} \in \mathbb{R}^{m\times d}$, and $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ are trainable weight matrices, and d is the dimension of the hidden layer.

3.4 Decoupled Temporal Attention Module

Construction of relative time matrix. For different users, in addition to general preferences for items, users tend to develop customized preferences based on their demand cycles. These preferences are primarily reflected in the position and timing information of items within a user's sequence of sets. Typically, the position of items in a set sequence includes both the absolute position of the item, the absolute position of the set it belongs to, and the temporal relationships of items relative to others, with items within the same set having a relative time of zero for each other.

In this context, we use the relative time between two items instead of the absolute time interval, because even if the time interval between two items is identical, the impact of a previously interacted item on a subsequent item may differ from the impact of a subsequent item on a previous one. The relative time between any two items v_i and v_j is defined as $r_i - r_j$. However, due to variations in user interaction frequency and the need to efficiently store relative time matrices, we compress the relative time data. For a user's interaction time sequence $\mathcal{R} = \{r_1, r_2, \ldots, r_L\}$, let $r_{min} = |r_m - r_n|$ represent the smallest absolute time difference between any items in the sequence. Thus, the relative time can be compressed into:

$$(r_i - r_j) \leftarrow \left\lceil \frac{|r_i - r_j|}{r_{min}} \right\rceil \times \frac{|r_i - r_j|}{r_i - r_j} \tag{7}$$

Assume k is the maximum compressed relative time between any two sets in the sequence. To ensure the non-negativity of time differences and to constrain them within a reasonable range for subsequent embedding retrieval operations, we map $r_i - r_j$ to the interval $[0, 2k]$ using the following formula:

$$\theta(i,j) = \begin{cases} r_i - r_j + k, & \text{if } -k \le r_i - r_j \le k \\ 0, & \text{if } r_i - r_j < -k \\ 2k, & \text{if } r_i - r_j > k \end{cases} \tag{8}$$

Furthermore, we construct a relative time matrix for each user's interaction sequence as follows:

$$\mathbf{T} = \begin{bmatrix} \theta(1,1), & \theta(1,2), & \ldots, & \theta(1,L^m) \\ \theta(2,1), & \theta(2,2), & \ldots, & \theta(2,L^m) \\ \vdots & \vdots & \ddots & \vdots \\ \theta(L^m,1), & \theta(L^m,2), & \ldots, & \theta(L^m,L) \end{bmatrix} \tag{9}$$

where L^m represents the maximum length of the item sequence, since each user's interactions are unique, the distribution of relative times also differs among users, making these matrices personalized. We initialize a learnable relative time embedding matrix $\mathbf{M}^{rt} \in \mathbb{R}^{2k \times d}$ for the items, where each mapped relative time in $\mathbf{T}$ can find its corresponding embedding in $\mathbf{M}^{rt}$.

Self-attention with multiple positional decoupling. The final position-decoupled attention involves the joint fusion of the absolute position of items, the absolute position of sets, and the relative time matrix. Given a hidden size d_h, a maximum sequence length l, a maximum set length s, an item embedding matrix $\mathbf{M}^{id} \in \mathbb{R}^{n \times d}$, an absolute item position embedding matrix $\mathbf{M}^{ip} \in \mathbb{R}^{l \times d}$, and an absolute set position embedding matrix $\mathbf{M}^{sp} \in \mathbb{R}^{s \times d}$, we have linear projections $\mathbf{W}_Q^{id}$, $\mathbf{W}_V^{id}$, $\mathbf{W}_K^{ip}$, $\mathbf{W}_K^{sp}$, $\mathbf{W}_K^{rp} \in \mathbb{R}^{d \times d}$, where the superscripts Q, K, and V represent queries, keys, and values, respectively. We have:

$$\mathbf{Q}^{id} = \mathbf{M}^{id}\mathbf{W}_Q^{id}, \quad \mathbf{K}^{ip} = \mathbf{M}^{ip}\mathbf{W}_K^{ip}, \quad \mathbf{K}^{sp} = \mathbf{M}^{sp}\mathbf{W}_K^{sp}, \quad \mathbf{K}^{rp} = \mathbf{M}^{rt}\mathbf{W}_K^{rp} \tag{10}$$

$$\mathbf{ATT}_{i,j}^{p} = \text{Fusion}\left(\frac{\mathbf{Q}_i^{id}\mathbf{K}_j^{ip^T}, \mathbf{Q}_i^{id}\mathbf{K}_j^{sp^T}, \mathbf{Q}_i^{id}\mathbf{K}^{rp^T}}{\sigma}\right) \tag{11}$$

where $\mathbf{ATT}_{i,j}^{p}$ is the fused position-decoupled cross-attention score from item i to item j. σ is the scaling factor used to prevent excessively large values. The fusion method is similar to that in [17], employing additive, concatenation, and gating mechanisms. Specifically, the scaling factor σ is $\sqrt{3d}$ for the additive method and $\sqrt{d}$ for other methods. Finally, we obtain the position output:

$$\mathbf{h}_{pos} = \text{softmax}(\mathbf{ATT}^{p})(\mathbf{M}^{id}\mathbf{W}_V^{id}) \tag{12}$$

To obtain the final customized feature representation for the user, we add the obtained $\mathbf{h}_{enh}$ and $\mathbf{h}_{pos}$ together, then apply a linear transformation before feeding them into a feedforward neural network. After stacking two layers in the feedforward network, we achieve the final output. The input to each attention layer is the output of the previous one, which can be expressed as:

$$\hat{\mathbf{h}} = \Gamma(\mathbf{W}^{P}(\mathbf{h}_{enh} + \mathbf{h}_{pos}) + \mathbf{b}^{P}) \tag{13}$$

where $\mathbf{W}^{P} \in \mathbb{R}^{d \times d}$ is a learnable linear projection, $\mathbf{b}^{P}$ is a bias term, and Γ represents the mapping of the two-layer feedforward network.

3.5 Model Learning

Training. To train the proposed model more efficiently, we apply four primary augmentation methods for each training sample: (1) randomly masking items within the input set sequence; (2) randomly masking an entire set within the sequence; (3) randomly swapping two sets within the sequence; and (4) reversing the order of sets in the sequence. Specifically, when performing swap and reverse operations, the corresponding times of the items need to be swapped or reversed as well. It is important to note that time should not be masked. The specific augmentation methods are as follows:

1. **Input**: $[(i_i,i_2,\, i_3),\, (i_4,\, i_5),\, (i_6)] \xrightarrow{mask} [(i_i,[mask]_1,\, i_3),\, ([mask]_2,\, i_5),\, (i_6)]$
   Label: $[mask]_1 = i_2$, $[mask]_2 = i_4$
2. **Input**: $[(i_i,i_2,\, i_3),\, (i_4,\, i_5),\, (i_6)] \xrightarrow{mask} [(i_i,i_2,\, i_3),\, ([mask]_1),\, (i_6)]$
   Label: $[mask]_1 = i_4, i_5$
3. **Input**: $[(i_i,i_2,\, i_3),\, (i_4,\, i_5),\, (i_6)] \xrightarrow{swap} [(i_i,i_2,\, i_3),\, (i_6),\, (i_4,\, i_5)]$
4. **Input**: $[(i_i,i_2,\, i_3),\, (i_4,\, i_5),\, (i_6)] \xrightarrow{reverse} [(i_6),\, (i_5,\, i_4),\, (i_3,\, i_2,\, i_1)]$

These methods allow us to generate more training samples, which not only help alleviate data sparsity but also better capture positional dependencies by altering the positions of items/sets. The hidden vector corresponding to the final masked item is used to perform a dot product operation with the item set, followed by a softmax function to obtain the user's preference for items. We use a cross-entropy loss function to train the model:

$$\mathcal{L} = -\frac{1}{|\mathcal{B}_{u,*}|} \sum_{v_* \in |\mathcal{B}_{u,*}|} \log \frac{\sum_{i=1}^{|\mathcal{B}_{u,*}|} \exp(\langle \mathbf{o}_i, *, \mathbf{e}_{i,*} \rangle)}{\sum_{j=1}^{n} \exp(\langle \mathbf{o}_j, *, \mathbf{e}_j \rangle)} \tag{14}$$

where $\mathcal{B}_{u,*}$ means a randomized mask of $\mathcal{B}_u$, $\mathbf{o}_*$ is the final output of the mask item or item set by Eq.(13), and $\mathbf{e}_{i,*}$ is the initial embedding of the masked item or set, $\langle \cdot \rangle$ means a dot product operation.

4 Experiments

In this section, we conduct extensive experiments on three real-world datasets to answer the following research questions.

- **RQ1:** Does CoANBR outperforms current state-of-the-art methods?
- **RQ2:** What is the performance impact of different components in the CoANBR?
- **RQ3:** How do different position embeddings and hyperparameters affect CoANBR?
- **RQ4:** How to explain the validity of CoANBR by attention visualization?

4.1 Experimental Settings

Dataset We evaluate CoANBR on three widely used datasets in NBR, namely TaFeng[3], Jingdong [35] and Beauty [20], which are collected from real-world online shopping platforms. The TaFeng dataset contains four months of supermarket transactions. The Jingdong dataset includes purchase records from the first three months of 2018. The Beauty dataset provides user interaction data from Amazon for the first six months of 2014. For the TaFeng dataset, we retain products that account for 90% of the occurrence frequency, while we keep 80% for the other two datasets. Following the setup in [5], we sort the datasets by timestamp, designating the last set as the test set, the second-to-last as the validation set, and the remaining sets for training. The statistics of the preprocessed datasets are shown in Table 1.

[3] `https://www.kaggle.com/datasets/chiranjivdas09/ta-feng-grocery-dataset2`

Table 1: Statistics of the pre-processed datasets.

Datasets	#Users	#Items	#Interactions	#Baskets	#Baskets/Users	#Interactions/Users
TaFeng	9,834	8,507	429,056	74,879	7.61	5.73
JingDong	3,279	4,656	26,900	21,016	6.41	1.28
Beauty	7,376	17,187	47,152	31,227	4.23	1.51

Evaluation Metrics. **Recall** and **NDCG** have been used extensively in previous work [22,5]. Recall represents the probability of being predicted true in a positive sample. NDCG assigns greater weight to higher-ranked items. We rank the top K items in terms of predicted probability and evaluate the performance of the model by Recall@K as well as NDCG@K with $K = 5, 10$ (i.e., R@5, R@10, N@5 and N@10).

Baselines and Implementation Details. We compare the proposed method with the following baseline methods: (1) Traditional methods: **PerPOP**, **FPMC** [24], and **TIFUKNN** [10]. (2) Deep learning methods: **DREAM** [33], **Sets2Sets** [9], **BART4Rec**[11], **CLEA** [22], **ETGNN** [35], **SFCNTSP** [34], and **MMNR** [5]. All compared methods are based on open-source code provided by the authors. The CoANBR has trained using the Adam optimizer, with the learning rate tuned in {0.01,0.005,0.001,0.0005} and model dimensions tuned in the range {32, 64, 128, 256, 512}. To prevent over-fitting, we use Dropout, and the dropout rate is directly adjusted at {0.1,0.2,0.3,0.4,0.5}. Models involving the self-attention approach all use a two-layer encoder and the number of attention heads is tuned in {2, 3, 4}. For the top-k nearest neighbors of the item, we fix the value to 10.

4.2 Overall Performance (RQ1)

We compared the proposed method with baselines on three publicly available datasets, the result is shown in Table 2. Our method is shown to outperform the baselines on all metrics. We obtain the following conclusions based on the experimental results: We compare our proposed method with baselines on three public datasets, achieving superior performance across all metrics, as shown in Table 2. We have findings as follows:

(1) Our proposed CoANBR outperforms other baselines across all three datasets. This success can be attributed to two main factors. Firstly, we enhance users' general preferences by leveraging collaborative relationships within sets. Secondly, we employ a decoupled positional encoder that captures interactions among elements in a set sequence by integrating absolute positions with relative timing, thereby enabling the customization of item preferences for users. (2) MMNR and CLEA each have their strengths, highlighting the importance of learning user preferences through item interactions. MMNR benefits from constructing differentiated representations for each interaction item, capturing a comprehensive understanding of user preferences. CLEA automatically extracts historical items relevant to the target item, mitigating noise in the sequence to better capture user preferences. (3) Sets2Sets generally performs better than

Table 2: The results on real-world datasets. **Red** denotes the best results and **underline** denotes the second-best results.

Method	TaFeng				JingDong				Beauty			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
PerPOP	8.07	11.74	8.41	10.28	26.59	28.72	22.40	23.17	0.76	1.45	0.47	0.71
FPMC	6.39	7.61	5.82	6.21	21.66	28.76	15.43	17.72	1.42	1.95	0.97	1.14
TIFUKNN	9.76	14.59	9.36	10.85	29.78	37.45	24.62	27.28	1.59	2.23	0.65	0.87
DREAM	9.16	11.12	8.81	9.80	23.08	29.43	17.24	19.34	1.85	2.87	1.28	1.62
Sets2Sets	9.39	13.75	9.59	11.19	27.75	32.99	23.35	25.19	2.14	3.47	1.64	2.08
BART4Rec	9.11	11.70	10.07	11.15	**<u>35.01</u>**	**<u>42.09</u>**	27.71	30.24	3.52	5.53	2.38	3.07
CLEA	10.95	14.97	10.97	**<u>13.12</u>**	29.78	38.07	24.69	28.82	**<u>3.73</u>**	**<u>5.58</u>**	**<u>2.65</u>**	3.11
ETGNN	9.23	12.08	10.05	11.42	30.64	37.20	25.00	27.22	3.61	5.67	2.53	3.25
SFCNTSP	10.56	11.53	10.03	11.32	31.89	38.12	25.31	27.82	3.65	5.41	2.54	3.29
MMNR	**<u>11.21</u>**	**<u>15.03</u>**	**<u>11.13</u>**	13.01	32.14	39.12	**<u>28.34</u>**	**<u>31.12</u>**	3.70	5.47	2.54	**<u>3.31</u>**
CoANSR	**11.63**	**15.47**	**11.65**	**13.47**	**39.36**	**47.27**	**31.21**	**34.23**	**4.33**	**6.54**	**2.76**	**3.55**
Improve(↑)	3.75%	2.93%	4.67%	2.67%	12.42%	12.31%	10.13%	9.99%	16.09%	17.20%	4.15%	7.25%

DERAM as it accounts for repeated elements in user behavior. TIFUKNN is particularly effective for datasets with multiple repeated items, underscoring the importance of considering repeated elements within set sequences. This is because items frequently appearing in the past are likely to reappear in the future. However, TIFUKNN performs poorly on more diverse datasets, such as Beauty, possibly due to its inadequate modeling of sequential dynamic information.

4.3 Ablation Study (RQ2)

To investigate the impact of each component on CoANBR, we conduct ablation experiments across three datasets. We have four variants of CoANBR: (1) "**w/o INA**" removes Item Neighboring Aggregation, disregarding the global information between items and sets. (2) "**w/o UPE**" removes the User Preference Enhancer, thus not enhancing users' general preferences. (3) "**w/o DTA**" removes the Decoupled Temporal Attention Module, ignoring customized user preferences. (4) "**w/o Rt**" removes the relative time attention within the Decoupled Temporal Attention Module, using only absolute positional information and item embeddings when computing attention.

Table 3: The ablation study of CoANSR.

Method	TaFeng		JingDong		Beauty	
	R@10	N@10	R@10	N@10	R@10	N@10
w/o INA	14.38	12.41	44.74	33.26	6.20	3.41
w/o UPE	14.01	11.48	44.62	32.89	6.14	3.33
w/o DTA	13.24	11.29	44.40	31.91	6.11	3.24
w/o Rt	14.56	11.90	46.49	33.38	6.38	3.38
CoANSR	**15.47**	**13.47**	**47.27**	**34.23**	**6.54**	**3.55**

As shown in Table 3, firstly, across all three datasets, there is a sharp decline in all metrics for w/o DTA. This is due to the direct addition of positional embedding to item embedding causing side effects such as information overwhelming[17], making it difficult to capture customized user preferences and also interfering with the collaborative information intended to enhance item embeddings, thus preventing optimal performance. Secondly, w/o Rt exhibits the best R@10 metric among the four variants, indicating that the combination of collaborative information and positional decoupling can significantly enhance performance. However, its performance in N@10 is inferior to w/o INA, which confirms the necessity of relative timing in capturing interactions between elements. Finally, omitting any component leads to a corresponding impact on performance, highlighting the importance of utilizing these components.

4.4 Detailed Performance Analysis (RQ3)

Impact of different kinds of positions. To understand the contribution of different positional information in the Decoupled Temporal Attention Module of CoANBR, we analyze various positional embeddings. We consider four types of positional embeddings: (1) Only Absolute (Only Abs), which includes the absolute position of items and the absolute position within the sequence. (2) Combining absolute position with sequence-relative position (Abs+SRP). (3) Combining absolute position with the time interval between items (Abs+TI). (4) Combining absolute position with the relative time of items (Abs+RI). The results are presented in Table 4.

Table 4: The results of using different position embedding on JingDong dataset.

Position Type	R@5	N@5	R@10	N@10
Only Abs	37.32	30.22	46.49	33.38
Abs+SRP	38.80	30.86	46.50	33.58
Abs+TI	38.99	31.01	47.09	33.89
Abs+RI	**39.36**	**31.21**	**47.27**	**34.23**

The introduction of relative positions improves performance, particularly in terms of R@5 and N@5, and the incorporation of relative time outperforms other methods across all metrics. This indicates that when introducing relative positions, one should consider not only the irregularities in time intervals between sequences in the recommendation task, but also the order of purchase times is equally important.

Impact of Hyperparameters. In this section, we investigate the impact of hyperparameters on performance. We perform experiments on the item similarity balancing parameter α and the item embedding size d, respectively. To be fair, we fix d to be 128 when experimenting with α, and we fix α to be 0.5 when experimenting with d. Figure 4 presents the performance of these two parameters on the JingDong dataset.

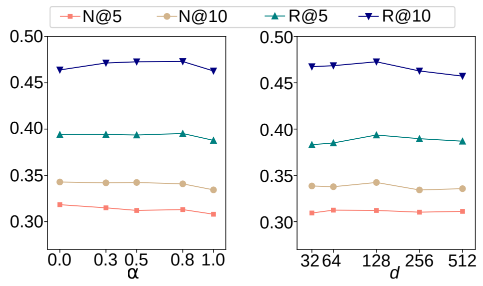


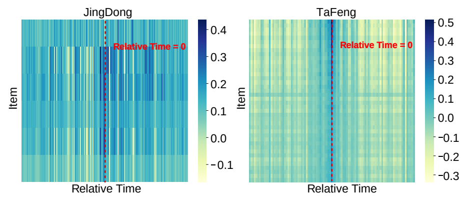
Fig. 4: Impact of hyperparameters: α and the dimension d on JingDong dataset.

– **Impact of α.** We can see that more stable and better performance is shown when $\alpha = 0.3$ and $\alpha = 0.8$, and performance is worst when α is 1, in other words, it cannot show good performance when only user-based collaboration information is used. Meanwhile, when α is 0, it produces unstable performance on Recall@10. It demonstrates the necessity of the balance of user-based and set-based collaborative information.

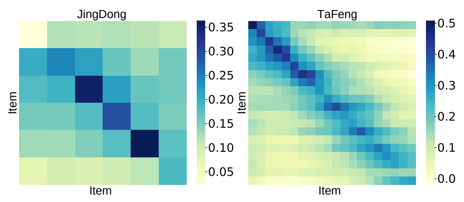
– **Impact of dimension** d. As the embedding size increases, the performance improves accordingly, and we achieve the best performance when d is 128. After that, the performance tends to decrease as the embedding size increases, the reason may be that our method aggregates the neighbor information, so too large dimensions can lead to overfitting.

4.5 Attention Visualization (RQ4)

In this section, we explain why CoANBR is effective by visualizing the attention matrix. We select some test samples on JingDong and TaFeng datasets, and the visualization result is shown in Figure 5. Figure 5a shows the average attention weights (i.e., $\mathbf{M}^{rt}$) for items at different relative times on the two datasets, with the absolute value of relative time increasing towards both ends, marked by a red line. Figure 5b represents the average attentional weights (i.e., $\mathbf{W}_K$) of different items in a sequence based on the same relative time. From Figure 5, we can draw the following conclusions: (1) Figure 5a indicates that the closer the relative time of interaction is to the target item, the greater the impact, suggesting that users' preferences change over time, and recent behaviors are more reflective of users' current interests. (2) The lack of complete symmetry in Figure 5b indicates that even with the same time interval, two items can have varying mutual influence due to users' diverse demands and interests.



(a) The average attention weights for items at different relative times.



(b) The average attentional weights of different items within a sequence.

Fig. 5: Relative Time-Based Attention Matrix Visualization on JingDong and TaFeng Datasets.

5 Related Work

5.1 Next Basket Recommendation

Next basket recommendation is extensively studied and becomes an integral part of business management, especially with the rise of e-commerce [21,29]. A significant amount of research proposes effective methods for the NBR problem. FMPC [24] introduces a personality transition graph based on matrix factorization and Markov chains, creating a transition matrix for each user to model sequential behaviors. Wang et al. [28] propose a hierarchical representation model (HRM), addressing FMPC's limitation of only using linear operations to capture multiple factors. However, both approaches rely on Markov chains and only capture local features between two adjacent baskets, failing to effectively model users' global dynamic features. Methods based on recurrent neural networks (RNN) [33] can alleviate this issue. Additionally, some works suggest using hierarchies to incorporate items' attribute information [2] or utilizing self-attention mechanisms to learn heterogeneous information separately for item IDs and attributes. The method most relevant to NBR is Sets2Sets [9], an RNN-based encoder-decoder framework for learning complex relationships between sets. However, some researchers [10] argue that RNNs cannot capture personalized item frequency (PIF) information, which plays a crucial role in guiding the NBR, especially in datasets with highly repetitive behavior. Thus, a k-nearest neighbor (KNN) method based on item occurrence frequency is proposed to directly utilize PIF co-signals. Many other studies make significant efforts in modeling item relevance in NBR tasks [22,27]. Although effective, most focus solely on the current sequences set, overlooking potential guidance from other users' interaction sequence sets.

5.2 Attention-Based Recommendation

The attention-based mechanism catalyzes research in sequential recommendation, demonstrating strong performance by assigning greater weight to key items in user interaction sequences. Inspired by the use of Transformers in NLP, SASRec [12] introduces a self-attention mechanism for ranking recommendations, adaptively assigning weights to preceding items at each step. Bert4Rec [26] identifies that the unidirectional structure of attention limits model representation and introduces deep bidirectional sequential modeling into recommendation systems. Others [6,39] propose designing attention modules for decoupling positions, considering only absolute positions. Unlike these, the relative positions in user interaction sequence sets are influenced by interaction timestamps and cannot simply assume equal spacing between sets. Moreover, during decoupled attention calculations, it is common to use only position-to-position self-attention [6,31], but with the introduction of relative time offers limited auxiliary information, making cross-attention between items and positions is crucial.

6 Conclusion

In this paper, we introduce a novel collaborative aggregation framework (CoANBR) for the next basket recommendation task, focusing on enhancing users' general preferences while customizing preferences for individual users. CoANBR consists of three modules: the Item-aware Aggregator and User Preference Enhancer, which capture users' general preferences by leveraging collaborative relationships between items and sets. The Decoupled Temporal Attention Module further refines these preferences by modeling the temporal dependencies of item position information within interaction sequences, enabling the customization of user preferences across different time contexts. The experiments on three real-world datasets demonstrate that our model outperforms other methods. For future work, we expect to expand our model to the larger scale of collaborative networks based on the collection of user profiles from real-world platforms.

Acknowledgments. We acknowledge that this study is supported by the National Natural Science Foundation of China under Grant No.62402105 and No. 62272098, the Central University Basic Research Fund of China under Grant No. 2242024K40019, the Major Project of Fundamental Research on Frontier Leading Technology of Jiangsu Province under Grant No. BK20222006, and Natural Science Foundation of Jiangsu Province under Grant No. BK20230815.

References

1. Ariannezhad, M., Jullien, S., Li, M., Fang, M., Schelter, S., de Rijke, M.: Recanet: A repeat consumption-aware neural network for next basket recommendation in grocery shopping. In: SIGIR. pp. 1240–1250 (2022)
2. Bai, T., Nie, J.Y., Zhao, Wayne Xin, e.: An attribute-aware neural attentive model for next basket recommendation. In: SIGIR. pp. 1201–1204 (2018)
3. Balasubramanian, K., Alshabanah, e.a.: Biased user history synthesis for personalized long-tail item recommendation. In: Recsys. pp. 189–199 (2024)
4. Choi, M., Kim, J., Lee, J., Shim, H., Lee, J.: Session-aware linear item-item models for session-based recommendation. In: WWW. pp. 2186–2197 (2021)
5. Deng, Z., Li, J., Guo, Z., Liu, W., Zou, L., Li, G.: Multi-view multi-aspect neural networks for next-basket recommendation. In: SIGIR. pp. 1283–1292 (2023)
6. Fan, X., Liu, Z., Lian, J., Zhao, W.X., Xie, X., Wen, J.R.: Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In: SIGIR. pp. 1733–1737 (2021)
7. Feng, C., Liang, J., Song, P., Wang, Z.: A fusion collaborative filtering method for sparse data in recommender systems. Information Sciences **521**, 365–379 (2020)
8. He, Y., Dong, Y., Cui, P., Jiao, Y., Wang, X., Liu, J., Yu, P.S.: Purify and generate: Learning faithful item-to-item graph from noisy user-item interaction behaviors. In: KDD. pp. 3002–3010 (2021)
9. Hu, H., He, X.: Sets2sets: Learning from sequential sets with neural networks. In: KDD. pp. 1491–1499 (2019)
10. Hu, H., He, X., Gao, J., Zhang, Z.L.: Modeling personalized item frequency information for next-basket recommendation. In: SIGIR. pp. 1071–1080 (2020)
11. Kang, T., Lee, H., Choe, B., Jung, K.: Entangled bidirectional encoder to autoregressive decoder for sequential recommendation. In: SIGIR. pp. 1657–1661 (2021)
12. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: ICDM. pp. 197–206. IEEE (2018)
13. Kumar, A., Hosanagar, K.: Measuring the value of recommendation links on product demand. Information Systems Research **30**(3), 819–838 (2019)
14. Li, J., Wang, Y., McAuley, J.: Time interval aware self-attention for sequential recommendation. In: WSDM. pp. 322–330 (2020)

15. Li, M., Liu, Yuanna, e.a.: Are we really achieving better beyond-accuracy performance in next basket recommendation? In: SIGIR. pp. 924–934 (2024)
16. Li, R., Zhang, L., Liu, G., Wu, J.: Next basket recommendation with intent-aware hypergraph adversarial network. In: SIGIR. pp. 1303–1312 (2023)
17. Liu, C., Li, Xiaoguang, e.a.: Noninvasive self-attention for side information fusion in sequential recommendation. In: AAAI. vol. 35, pp. 4249–4256 (2021)
18. Liu, W., Zhang, Yin, e.a.: Item relationship graph neural networks for e-commerce. IEEE TNNLS **33**(9), 4785–4799 (2021)
19. Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X.: Ultragcn: ultra simplification of graph convolutional networks for recommendation. In: CIKM (2021)
20. McAuley, J., Targett, e.a.: Image-based recommendations on styles and substitutes. In: SIGIR. pp. 43–52 (2015)
21. Morsy, S., Karypis, G.: Cumulative Knowledge-based Regression Models for Next-term Grade Prediction, p. 552–560 (Jun 2017)
22. Qin, Y., Wang, Pengfei, e.a.: The world is binary: Contrastive learning for denoising next basket recommendation. In: SIGIR. pp. 859–868 (2021)
23. Rashed, A., Elsayed, S., Schmidt-Thieme, L.: Context and attribute-aware sequential recommendation via cross-attention. In: RecSys. pp. 71–80 (2022)
24. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized markov chains for next-basket recommendation. In: WWW. pp. 811–820 (2010)
25. Seol, J., Gang, M., Lee, S.g., Park, J.: Proxy-based item representation for attribute and context-aware recommendation. In: WSDM. pp. 616–625 (2024)
26. Sun, F., Liu, J., Wu, Jian, e.a.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: CIKM. pp. 1441–1450 (2019)
27. Sun, W., Xie, R., Zhang, J., Zhao, W.X., Lin, L., Wen, J.R.: Generative next-basket recommendation. In: RecSys. pp. 737–743 (2023)
28. Wang, P., Guo, J., Lan, Y., Xu, J., Wan, S., Cheng, X.: Learning hierarchical representation model for nextbasket recommendation. In: SIGIR. pp. 403–412 (2015)
29. Wu, Yuxia, e.a.: Personalized long- and short-term preference learning for next poi recommendation. IEEE TKDE **34**(4), 1944–1957 (2022)
30. Wu, L., He, X., Wang, X., Zhang, K., Wang, M.: A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation. IEEE TKDE **35**(5), 4425–4445 (2022)
31. Xie, Y., Zhou, P., Kim, S.: Decoupled side information fusion for sequential recommendation. In: SIGIR. pp. 1611–1621 (2022)
32. Xin, X., He, Xiangnan, e.a.: Relational collaborative filtering: Modeling multiple item relations for recommendation. In: SIGIR. pp. 125–134 (2019)
33. Yu, F., Liu, Q., Wu, S., Wang, L., Tan, T.: A dynamic recurrent model for next basket recommendation. In: SIGIR. pp. 729–732 (2016)
34. Yu, L., Liu, Z., Zhu, T., Sun, L., Du, B., Lv, W.: Predicting temporal sets with simplified fully connected networks. In: AAAI. vol. 37, pp. 4835–4844 (2023)
35. Yu, L., Wu, G., Sun, L., Du, B., Lv, W.: Element-guided temporal graph representation learning for temporal sets prediction. In: WWW. pp. 1902–1913 (2022)
36. Yu, S., Ma, L., Gao, X., Guo, J., Chen, G.: Attentive hawkes process application for sequential recommendation. In: DASFAA. pp. 473–488. Springer (2023)
37. Zhang, Y., Cheng, D.Z.e.a.: A model of two tales: Dual transfer learning framework for improved long-tail item recommendation. In: WWW. pp. 2220–2231 (2021)
38. Zhou, K., Yu, H., Zhao, W.X., Wen, J.R.: Filter-enhanced mlp is all you need for sequential recommendation. In: WWW. pp. 2388–2399 (2022)
39. Zhu, Y., Huang, Bo, e.a.: Progressive self-attention network with unsymmetrical positional encoding for sequential recommendation. In: SIGIR (2022)