Modeling Temporal Positive and Negative Excitation for Sequential Recommendation

Chengkai Huang*
chengkai.huang1@unsw.edu.au
The University of New South Wales
Sydney, NSW, Australia

Shoujin Wang
shoujin.wang@uts.edu.au
University of Technology Sydney
Sydney, NSW, Australia

Xianzhi Wang
XIANZHI.WANG@uts.edu.au
University of Technology Sydney
Sydney, NSW, Australia

Lina Yao
lina.yao@unsw.edu.au
CSIRO's Data 61 and UNSW
Sydney, NSW, Australia

ABSTRACT

Sequential recommendation aims to predict the next item which interests users via modeling their interest in items over time. Most of the existing works on sequential recommendation model users' dynamic interest in specific items while overlooking users' static interest revealed by some static attribute information of items, e.g., category, brand. Moreover, existing works often only consider the positive excitation of a user's historical interactions on his/her next choice on candidate items while ignoring the commonly existing negative excitation, resulting in insufficiently modeling dynamic interest. The overlook of static interest and negative excitation will lead to incomplete interest modeling and thus impedes the recommendation performance. To this end, in this paper, we propose modeling both static interest and negative excitation for dynamic interest to further improve the recommendation performance. Accordingly, we design a novel Static-Dynamic Interest Learning (SDIL) framework featured with a novel Temporal Positive and Negative Excitation Modeling (TPNE) module for accurate sequential recommendation. TPNE is specially designed for comprehensively modeling dynamic interest based on temporal positive and negative excitation learning. Extensive experiments on three real-world datasets show that SDIL can effectively capture both static and dynamic interest and outperforms state-of-the-art baselines.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Sequential Recommendation, Temporal Information Modeling, Attention Mechanism

1 INTRODUCTION

Sequential recommender systems (SRSs) aim to generate sequential recommendations by predicting the next item which interests a given user. SRSs generally model the user's dynamic and timely interest in items from a sequence of historically interacted items for the recommendation. Benefiting from the strength of well-capturing users' interest changes over time, SRSs are able to provide more accurate and timely recommendation results to users [30].
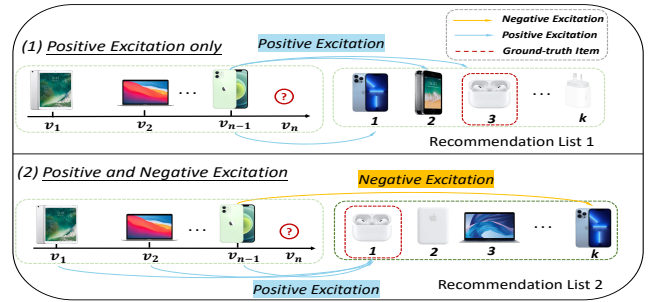

Figure 1: An example of recommendations via modeling positive excitation only (existing methods) and modeling both positive and negative excitation (our proposal). Clearly, the latter achieves better performance via ranking the ground-truth next item AirPods at the Top-1 position in the recommendation list.

A variety of SRS methods based on different models including Markov chain models, latent representation learning models, and deep learning models have been proposed in recent years. For instance, Markov chain-based SRSs adopt Markov chain models to model the first-order transitions over user-item interactions within an interaction sequence to predict the probable next item [21, 23]. Distributed representation learning-based methods map each interaction into a latent representation via capturing the contextual information for next-item recommendations [12]. In recent years, advanced models including shallow or deep neural models have been employed in SRSs to improve the recommendation performance. For example, Recurrent Neural Networks (RNN) have been utilized to model the long- and short-term sequential dependencies in a sequence of interactions for next-item recommendations [10]. Convolutional Neural Network (CNN) [36] and self-attention [14] models have been incorporated into SRSs for capturing more complex sequential dependencies (e.g., skip dependencies or collective dependencies) for next-item recommendations. These SRS methods have achieved remarkable recommendation performance.

Although remarkable recommendation performance has been achieved, there are still some significant gaps which prevent the further performance improvement of existing SRS methods. To be specific, *most of the existing SRSs only model users' dynamic interest in items while overlooking users' static interest [38]* (**Gap 1**).

---

*Corresponding author.

However, static interest is also an important factor to determine which item to interact with for a given user. In this paper, **dynamic interest** means users' interest in specific items, which are captured from the ID information of a sequence of items interacted by each user. For instance, a user may be interested in iPhone 12. Item ID often changes frequently in most sequences, leading to frequent fluctuations of users' dynamic interest. **Static interest** refer to users' interest at a relatively higher level (e.g., item category or brand) and are usually more stable and change less frequently. For instance, a user may be interested in Apple products. Obviously, static interest changes less frequently than dynamic interest.

More importantly, *most of the existing SRSs cannot thoroughly capture users' dynamic interest since they often only model the positive excitation while overlooking negative one* (**Gap 2**). In SRSs, the **positive** (resp., **negative**) **excitation** refers to one prior interaction has a positive (resp., negative) impact on the next interaction. For instance, as an individual user, Alice may purchase a lens after she has purchased a Canon camera. However, it is unlikely that she will purchase another Nikon camera in a short period. In such a case, the purchase of Canon camera has positive excitation on the purchase of lens and negative excitation on that of Nikon camera. Clearly, negative excitation is an important signal to indicate which item may not be of the user's interest at the next time point and thus is necessary for accurate sequential recommendations. Although a few works in the literature have tried to model excitation for sequential recommendations, they generally only consider the positive excitation while ignoring the negative one [26] [31]. For example, Wang et al. [26] defined a short-term and life-long positive self-excitation function for next-item recommendations.

Aiming at bridging the aforementioned two significant gaps in exiting SRSs, we propose a novel **S**tatic-**D**ynamic **I**nterest **L**earning (**SDIL**) framework. SDIL is able to comprehensively model users' static and dynamic interest in items for generating accurate sequential recommendations. SDIL consists of (1) a Static Interest Modeling (SIM) module, (2) a Dynamic Interest Modeling (DIM) module, and (3) a next-item prediction module. To be specific, given a sequence of items interacted by a user, the corresponding attribute information of these items is input into the SIM module to capture the static interest of the user. Meanwhile, the ID information of these items is imported into the DIM module to comprehensively capture the user's dynamic interest. Then, both types of interest are well integrated as the input of the downstream prediction module for the next-item prediction.

More importantly, we devise a novel **T**emporal **P**ositive and **N**egative **E**xcitation Modeling (TPNE) framework to quip the DIM module, which constitutes the main contribution of this work. Inspired by [26], TPNE is built on the basis of the temporal point process (TPP) [22], which is a classical model for modeling discrete event sequences in a continuous time period. In TPNE, first, a relation-based temporal module is designed to model the positive or negative excitation of each of a user's historical interactions on his/her next choice on candidate items. Here, the relation mainly refers to the substitute/complementary relations between items, which are extracted from the "co-click" (of items), "co-purchase" data. Then, a novel time decay kernel function is particularly designed to measure the excitation strength of each historical interaction according to the time interval between it and the next choice.

Time interval is a key factor to determine whether the excitation strength is strong or weak. [26, 31]

The main contributions of this work can be summarized below:

- We propose a novel Static-Dynamic Interest Learning (SDIL) framework for comprehensively modeling users' interest in items. SDIL consists of a static interest modeling (SIM) module, a dynamic interest modeling (DIM) module, and a next-item prediction module.
- We devise a novel Temporal Positive and Negative Excitation Modeling (TPNE) framework to quip the DIM module. TPNE is good at modeling both the positive and negative excitation as well as the excitation strength.
- We evaluate our SDIL on three real-world datasets with different characteristics. Extensive experiments not only show the consistent superiority of SDIL over state-of-the-art baselines but also verify the rationality and effectiveness of our design in SDIL.

2 RELATED WORK

2.1 Sequential Recommendation

Sequential recommendation intends to recommend items that may interest users by modeling the sequential dependencies across users' historical interactions. A variety of methods have been proposed for the SRS task including rule-based methods [34], KNN-based methods [13], Factorization machine-based methods, and Markov chain-based methods [7, 21]. Nevertheless, these methods fail to model the long user behavior sequences in terms of the limited representation ability of models. Recent years have witnessed the success of deep learning-based methods applying in user modeling [8, 9], various models like CNN-based model [36], RNN-based models [10] [11], self-attention based models [14] [37] were explored for SRS task. We briefly review attention-based SRS methods, which are most relevant to our work. The self-attention mechanism was proposed by Vaswani et al. [24]. Benefiting from its strong ability to model the correlations among context information, many works employed it in SRS by intensifying the correlation between important historical items and target items. Li et al. proposed the NARM [16], which leverages a global and local encoder with an attention module to model both user short- and long-term interest. Wang et al. [29] propose the ATEM, which is an attention-based transition embedding model. It builds the attentive context embedding without order assumption. Kang et al. [14] utilizes the transformer architecture to bridge the item-item relationship, which gains significant performance in the next-item prediction task. However, all of these methods only focus on the item relations s within sequences, overlooking the vital temporal signals, which might lead to the sequential patterns being hard to generalize on unseen timestamps or different time intervals.

2.2 Time-sensitive Recommendation

Temporal recommendation considers the temporal evolution of user interest in item transition, represented by methods based on matrix factorization. TimeSVD++ [15] achieved strong results by splitting time into several bins of segments and combining it into the CF framework. Bayesian Probabilistic Tensor Factorization(BPTF) [33] is proposed to include time as an extra dimension for the tensor

factorization, which is an extension of traditional MF methods. After that, a variety of works attempt to model the long- and short-term user interest separately. For instance, Zhu et. al. [40] propose the TimeLSTM to capture users' long and short-term dynamic interest through the specific time gates. Cai et. al. [1] propose the LSHP, combining the short-term and long-term Hawkes Processes based on different segments of user history to predict the type and time of the next action in sequential online interactive behavior modeling. Wang et. al. [26] also leverages the Hawkes Processes to model the user repeat consumption behavior from the perspective of short- and life-long terms. After that, Wang et al. [27] designs the multiple forms of temporal decay functions for item relations in the user interaction history. Recently, Wang et al. [25] have devised relational intensity and frequency domain embeddings to adaptively determine the importance of historical interactions. Besides, Fan et.al. [5] design a new framework TGSRec upon a pre-defined continuous-time bipartite graph, which can capture collaborative signals from both users and items, as well as consider temporal dynamics within sequential patterns. Although these time-sensitive methods well model the user's dynamic interest changes, they do not explicitly model the user's relatively stable interest. Meanwhile, most methods attempt to learn the positive temporal collaborative signals. but ignoring the effectiveness of negative temporal signals.

3 PRELIMINARY

3.1 Problem Formulation

Let $\mathcal{U} = \{u_1, u_2, ..., u_{|U|}\}$ and $\mathcal{V} = \{v_1, v_2, ..., v_{|V|}\}$ be user set and item set respectively. For each user $u \in \mathcal{U}$, there is a chronological sequence of items interacted by u, denoted as $H_u = \{v_1, v_2, ..., v_n\}$, where n is the length of the sequence H_u. For each item $v_i \in H_u$, it is associated with a set of attribute information including item category c_i, item brand b_i, item price p_i and interaction timestamp t_i. Generally, for a user u, given her/his sequence context $O = \{v_1, v_2, ..., v_{n-1}\}$ together with the associated item attribute information, the task of an SRS is to predict the next item (target item) v_n which may interest u.

3.2 Hawkes Processes in Sequential Modeling

Hawkes Processes is one of the most classic evolutionary processes [3, 6]. The traditional Hawkes Processes is defined by the following conditional intensity function:

$$\lambda(t) = \mu(t) + \sum_{t_i < t} \varphi(t - t_i), \tag{1}$$

where $\mu(t)$ represents the base intensity of the model. t and t_i denote the happening time of the target event and that of the last event respectively. Hence, $t - t_i$ denotes the time interval between the target event and the last event. $\varphi(\cdot)$ denotes the exciting function of a Hawkes Process.

In practice, in the sequential recommendation scenario, a user's interest changes over time, and thus the interest evolution process can be formalized as a Hawkes Process. Accordingly, the first term $\mu(t)$ in Eq (1) can be used to indicate the basic user interest in a target item v_n. The second term in Eq (1) means the accumulative

impact of all the historical context items bought before time t on the user's interest at time t.

4 METHODOLOGY

4.1 Overview

The framework of our proposed model is shown in Figure 2. Our model is mainly composed of (1) a temporal dynamic user interest modeling module, (2) a static user interest modeling module, and (3) a next-item prediction module. The temporal dynamic user interest modeling module models users' dynamic interest which changes over time by taking the ID information of historically interacted items as the input. Specifically, both the positive and negative excitation of each historical interaction on the user's next choice will be carefully considered. In addition, the temporal information (i.e., the time interval between a historical interaction and the next interaction) is well utilized to measure how positive or negative the excitation is. The static user interest modeling module models users' static interest in items based on static attribute information of items (e.g., item category, item brand) which are interacted by each user. Finally, both the learned dynamic interest and static interest are well integrated into the prediction module for the recommendation of the next item.

4.2 Users' Dynamic Interest Modeling

Temporal point processes (TPPs) are a flexible and powerful paradigm that is good at modeling discrete event sequences localized in a continuous time period. A sequence of items interacted by a given user in a period well falls into this scenario. Hence, it is natural to take a TPP as the basic structure to model user-item interaction sequences. However, a classical TPP usually considers the positive excitation of each prior event (e.g., user-item interaction) on the following event (e.g., a user's next choice) and cannot model the possible negative excitation. Hence, they cannot be directly utilized to model both positive and/or negative excitation of a user's historical interactions on his/her next choice. To this end, we devise a novel Temporal Positive and Negative Excitation learning (TPNE) module to model both positive and negative excitation of historical interactions to well capture the user's dynamic interest. More importantly, TPNE is able to specify the excitation strength according to the happening time of each historical interaction.

To be specific, we take the TPP approach as a base architecture to build TPNE for capturing users' dynamic interest. The most classic model for the TPP is the Hawkes process, which is formulated as:

$$\lambda_T(t) = \lambda_0 + \sum_{i:t_i < t_n} \varphi(t_n - t_i), \tag{2}$$

where λ_0 denotes the user's basic interest in a candidate item. $\sum_{i:t_i < t_n} \varphi(t_n - t_i)$ denotes the historical user-item interaction's excitation on target item v_n. For the first term, it can be calculated as:

$$\lambda_0 = e_h^T e_{v_n} + u_b + i_b, \tag{3}$$

where e_h denotes the user historical interest embedding revealed from its historically interacted items and e_{v_n} denotes the target item ID embeddings. u_b and i_b denote the user bias and item bias respectively. To well model the relationship between e_h and e_{v_n}, we adopt the self-attention module to train the basic interest value λ_0.
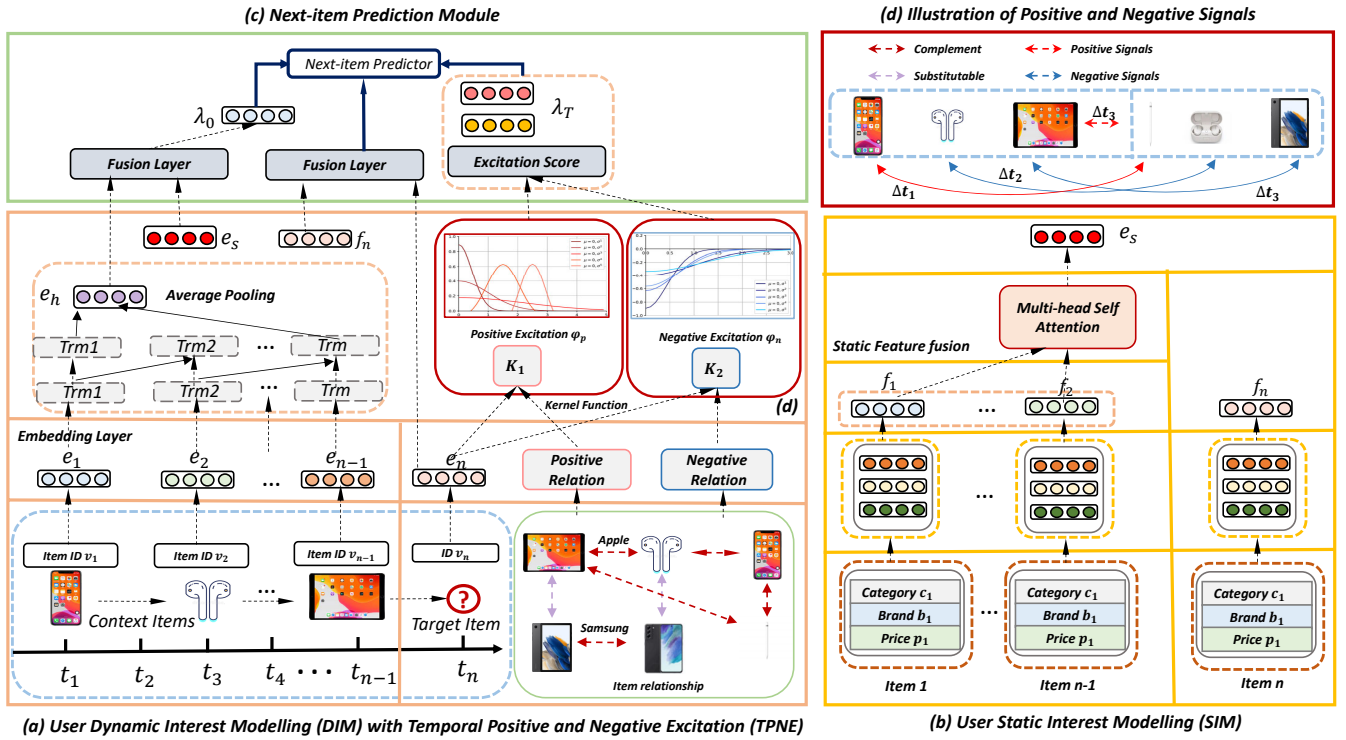
Figure 2: The framework of our proposed SDIL framework. SDIL is composed of three main modules: user dynamic interest modeling module (DIM), user static interest modeling module (SIM), and next-item prediction module. DIM captures a user's evolutionary dynamic interest by carefully modeling the time-sensitive positive and negative excitation of the user's historical interactions on the user's current interest. The SIM captures the user's relatively stable and high-level interest (e.g., interest in item category, brand) from the attribute information of interacted items. Finally, the prediction module well integrates both dynamic and static interest to obtain more precise interest for next-item prediction.

Given a user u, the input sequence is made up of item IDs sequence $H_u = \{v_1, v_2, ..., v_n\}$. To obtain a unique dense embedding for each item ID, we use a linear embedding layer. Then we can get the input item embedding matrix $E \in \mathrm{R}^{|V| \times d}$ ($|V|$ denotes the number of items and d denotes the dimension of item embeddings). For each item, its embedding $e_v = E(v) \in \mathrm{R}^{1 \times d}$. Then we can input H_u's item IDs to E and obtain the user u's item embedding sequences and represent it as a matrix $E(H_u)$. Then, following previous works [14, 31, 32], we introduce the position embedding pos_i in each timestamp so as to add the position information of historical items. The input are item ID embeddings' matrix $E(H_u)$ and corresponding position embeddings POS_i:

$$A_i = Att((E(H_u)+POS_i)W^Q, (E(H_u)+POS_i)W^K, (E(H_u)+POS_i)W^V),$$
(4)

$$Att_i = Attention(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$
(5)

where Q, K, and V denote query, key, and value respectively. W^Q, W^K and W^V denote the linear transformation matrices.

$$E(H_u) = LayerNorm(H_u + Dropout(FFN(A_i))).$$
(6)

Similar to previous works [31, 32], we introduce the LayerNorm, Dropout, and FFN layers to alleviate the over-fitting problem. Finally, we adopt the average pooling operation to get the user historical interest embedding e_h:

$$e_h = \frac{1}{|N| - 1} \sum_{i=1}^{n-1} E(H_u)_i,$$
(7)

where $|N|$ denotes the length of the user u sequence. Then we can calculate the the first term λ_T ($e_{v_n} = E(v_n)$) as Eq. (3).

For the second term, it denotes the multi-excitation process, which represents the user history items' impact on the target item, that is to say, the dynamic user interest in target item v_n. Thus, it is important to curve the dynamic temporal kernel function.

Current Hawkes Process or Poisson process modeling states that all past events should have positive influences on the occurrence of current events. However, in real-world applications, there are many situations that past events give negative effects on current events. For example, after a user bought a new iPhone, in the short-term, this purchase action would give a negative excitation signal to other brand mobile phones. Inspired by previous work[26][27][4], we propose modeling both positive and negative excitation for SRS task, which is composed of two parts **Positive Excitation**

Learning and **Negative Excitation Learning**. where φ_p denotes the positive factor, and φ_n denotes the negative factor. We leverage both positive and negative signals together, making up the reactive point processes:

$$\lambda_T(t) = \lambda_0 + \sum_{i:t_i < t_n} \varphi_p(t_n - t_i) - \sum_{j:t_j < t_n} \varphi_n(t_n - t_j), \qquad (8)$$

where λ_0 denotes the basic preference of the user on item v_n as previously mentioned, t_i denotes the happening time of positive interaction related to v_n. In contrast, t_j denotes the happening time of negative interaction related to v_n. Then we will discuss them separately.

*4.2.1 **Positive Excitation Learning**.* For seeking the positive correlation between historical items O and target item v_n, we introduce four different explicit relations: also_buy (r_1), also_view (r_2), share_brand (r_3) and similar_item (r_4) contained in the datasets. For also_buy and share_brand relations, we regard them as complementary relations. While, for also_view and similar_item(two items have a similar price within the same category), we regard them as substitute relations. Then positive excitation learning can be defined as:

$$\varphi_p(t_n - t_i) = \sum_{i:t_i < t} I_{rp}(v_i, v_n)\mathcal{K}_1(t_n - t_i), \qquad (9)$$

where I_{rp} denotes the indicator function, if historical item v_i has a relation $r \in \{r1, r2, r3, r4\}$ with v_n, then the I_{rp}=1, otherwise 0. It bridges the excitation between context information and target user interest. $\mathcal{K}_1(\cdot)$ denotes the positive temporal kernel function, which is composed of two parts as follow:

$$\mathcal{K}_1^i(\Delta t_1) = N\left(\Delta t_1 \mid 0, \sigma_1^v\right) + N\left(\Delta t_1 \mid \mu_2^v, \sigma_2^v\right). \qquad (10)$$

Previous works [26, 31] commonly model the short-term positive effect of historical items on target item, which only leverages the complementary information($r1$ and $r3$) to obtain the positive excitation. We also model this positive complementary relation signal in the first term $N\left(\Delta t_1 \mid 0, \sigma_1^v\right)$, ($\Delta t_1 = t_n - t_i$ represents the time interval between the target item v_n and the related historical item v_i). σ_1^v is decided by item ID v_i, which is item-specific parameter. We leverage a normal distribution function to simulate the user dynamic interest changes. For example, if a user buys a mobile phone, who may have a high possibility to buy accessories for the mobile phone in a short-term window, such as mobile phone film, matching earphones and so on. However, with the growth of time and the aging of the mobile phone, the interest of users to buy the corresponding supporting products would gradually decrease. Thus, we choose a mean equals to 0 normal distribution to model the user interest decay process. For the second term, which happens because many items purchased by users have corresponding lifespans. When the product life cycle is just approaching, the probability of users purchasing similar substitute products will be a strong positive incentive. For example, if a user buys a mobile phone, and he/she needs to replace a brand new mobile phone one year later, then this will bring a positive excitation for the substitute relationship of similar items. We model this temporal relationship using a normal distribution with mean μ_2^v, which is also related to v_i, which assumes the user begins to buy the substitute products after μ_2^v time interval.

*4.2.2 **Negative Excitation Learning**.* On the other hand, for seeking the negative correlation between historical items O and target item v_n, we leverage two different explicit relations: also_view (r_2) and similar_item (r_4) contained in the datasets. We treat them as substitute relations and define the negative excitation learning as:

$$\varphi_n(t_n - t_j) = \sum_{j:t_j < t_n} I_{rn}(v_j, v_n)\mathcal{K}_2(t_n - t_j), \qquad (11)$$

where I_{rn} denotes the indicator function, if historical item v_j has a relation $r \in \{r2, r4\}$ with v_n, then the I_{rn}=1, otherwise 0. $\mathcal{K}_2(\cdot)$ denotes the negative temporal kernel function:

$$\mathcal{K}_2(\Delta t_2) = -N\left(\Delta t_2 \mid 0, \sigma_3^v\right), \qquad (12)$$

where $N\left(\Delta t_2 \mid 0, \sigma_{s3}^v\right)$ is a normal distribution of $\Delta t(\Delta t_2 = t_n - t_j)$ with 0 mean and σ_3^v standard deviation. σ_{s3}^v is also related to v_j. From an empirical analysis, users rarely buy a large number of substitutes of the same category in a short period of time. For example, a simple example, a user has just purchased a Mac laptop, so the probability of him/her buying another Lenovo laptop in the short term would be very low. However, with time goes by, the negative effect would decrease gradually, so we use a negative normal distribution with $\mu = 0$ to fit this process. (Noting that μ_2^v, σ_1^v, σ_2^v and σ_3^v are all learn-able parameters.)

In summary, two different temporal patterns of SRS are modeled positive excitation learning processes and negative excitation learning processes. Since these two important temporal patterns occur frequently across the whole user-item interaction history, they can provide extra-temporal information to help us capture the dynamics of users' interest.

4.3 Users' Static Interest Modeling

In real-world scenarios, users also have relatively fixed preferences in sequential decision-making, such as high loyalty to a certain brand and the choice of the item price, which are relatively stable in the long term and not easy to change. Therefore, we believe that this part of the more stable user interest should also be included in the modeling to achieve more accurate recommendations.

To represent the users' relatively stable interest in their interaction history, we adopt the feature-based self-attention module to model the user's static interest. Because item IDs do not clearly indicate fine-grained attributes that users are interested in, we leverage the feature-based self-attention block to search for users' preferences on static feature-level patterns. Specifically, we project the discrete and heterogeneous attributes of items into low-dimensional dense vectors and fuse these vectors, then use multi-head self-attention to model the user's static interest. Without loss of generality, we choose the category, brand, and price of three different side information to comprehensively represent different aspects of one item. We also use a linear embedding layer to obtain category embedding matrix $C \in R^{|C| \times d}$, brand embedding matrix and $B \in R^{|B| \times d}$ price embedding matrix $P \in R^{|P| \times d}$ ($|C|, |B|, |P|$ denotes the number of categories, brands and price bins respectively). Furthermore, the $c_i \in R^{1 \times d}$, $b_i \in R^{1 \times d}$ and $p_i \in R^{1 \times d}$ to denote each item different feature embeddings respectively. Our approach is easily extended to include more features. We adopt a simple yet

effective additive feature fusion mode:

$$f_i = c_i + b_i + p_i, \tag{13}$$

where $f_i \in F$ denotes the fused feature-level item v_i representation, F denotes the stacked fusion matrices. Then for each user u, we can get his/her fused feature sequence $\{f_1, f_2, f_3, ..., f_n\}$. We utilize the multi-head self-attention network[24] to model the user's historic static interest. In the feature-level self-attention module, the query(Q), key(K), and value(V) are the same and equal to F, and we use three different projection transformation matrices to convert them into the same subspace:

$$H_f = Att(FW^Q, FW^K, FW^V). \tag{14}$$

Noting that we do not use the position here because we assume that user static interest is relatively stable and independent of user-item interaction timestamp. Where W^Q, W^K, and W^V are the linear transformation matrices. Then we adopt the multi-head attention mechanism to gain more semantic information from different subspaces.

$$M_f = Multihead(F) = Concat(h_1, h_2, ..., h_{l_f})W^O, \tag{15}$$

$$h_i = Att(FW_i^Q, FW_i^K, FW_i^V), \tag{16}$$

where l_f denotes the length of different heads, and here we also apply the layer normalization, FFN layers and residual connection module following previous works [38][32], alleviating the over-fitting problem in the training phase.

$$M_f = LayerNorm(M_f + H_f), \tag{17}$$

$$FFN(M_f) = ReLU(M_f W_1 + b_1)W_2 + b_2, \tag{18}$$

where W_1, W_2 and b_1, b_2 denotes the learnable parameters, we sum pooling all the h_i in each user sequence to obtain the user static interest representation e_s.

$$e_s = \frac{1}{|N| - 1} \sum_{i=1}^{|N| - 1} H_f, \tag{19}$$

e_s provides more stable user interest on certain attributes (i.e. brand, price in our case), such characteristics may not be easily perceived but are indeed potential consumption habits of users.

4.4 Next-item Prediction

Since both dynamic user interest and static user interest are vital in SRS task, it is crucial to combine them and as the input of the downstream prediction module for next-item prediction. Hence, inspired by previous works [18, 35], we leverage the gate fusion module to fuse the dynamic embedding e_h and static interest embedding e_s so as to better next-item prediction.

$$g = \sigma(W_1 e_s + W_2 e_h + b), \tag{20}$$

$$e_f = g \odot e_s + (1 - g) \odot e_h, \tag{21}$$

where W_1, W_2, and b are the learnable parameters in the gating layer. $\odot$ denotes the element-wise multiplication and $g \in \mathcal{R}^{1 \times d}$ denotes the learnable gate. Since there is some redundant information between user dynamic interest and static interest, we adaptively pass the informative messages and restrain the useless ones to obtain the final interest representation e_f. In order to learn the parameters

Table 1: Statistics of the datasets after preprocessing.

Specs.	Beauty	Cellphones	Toys
# Users	22,363	27,879	19,412
# Items	12,101	10,429	11,924
# Avg. Seq Length	8.8	7.0	8.6
# Interactions	198,502	194,439	167,597
# Sparsity	99.93%	99.94%	99.93%
# Item Categories	148	20	145

of the SDIL framework, we adopt the pairwise ranking loss (BPR loss) [20] to optimize our model:

$$\mathcal{L}_r = - \sum_{u \in \mathcal{U}} \sum_{i=1}^{N_u} \log \sigma \left(\hat{y}_{ui} - \hat{y}_{uj} \right), \tag{22}$$

$$\hat{y}_{ui} = e_f^T e_i + \lambda_{T,i}, \quad \hat{y}_{uj} = e_f^T e_j + \lambda_{T,j} \tag{23}$$

where σ represents the sigmoid function, $\hat{y}_{ui}$ represents the final preference score of user u to positive item i while $\hat{y}_{uj}$ represents the final preference score of user u to negative item j.

5 EXPERIMENTS

5.1 Experimental Setting

5.1.1 Dataset Preparation. We conduct experiments on publicly accessible Amazon datasets [19]. We choose three representative sub-datasets from Amazon datasets: *Beauty, Cell Phones and Accessories* (Cellphones) and *Toys and Games* (Toys) and keep the '5-core' datasets [39][21][2], which filter out user-item interaction sequences with length less than 5. The statistics of evaluation datasets are shown in Figure 1. Following the previous works[27][26], we take "also buy" as complementary relations and "also view" as substitute relations between items, i.e., items v_1 and v_2 are complementary if most users who buy v_1 also buy v_2. In addition, we introduce two more effective item relationships: (1) items of the same brand have complementary relations (e.g., iPhone and AirPod), and (2) items of the same category with similar prices have substitute relation (e.g., Canon cameras and Nikon cameras). Furthermore, we also take some item attribute information into account, including fine-grained item category, item brand and item price. Item category and brand are categorical features and we use the unique one-hot encoding to represent them. For price information, we bin and cut them into 10 intervals.

For reproducibility, we follow the commonly used benchmark setting of ReChorus [26, 27] to set up our experiments. Specifically, for each user, we first discard duplicated interaction sequences and sort the items in each user's sequence chronologically by their timestamp. Furthermore, the maximum length of interaction sequences is set to 20. If there are more than 20 interactions in a sequence, we adopt the latest 20 interactions. If the number of interactions in a sequence is less than 20, we make it up to 20 by padding virtual items with the ID of 0. For all baselines on all three experimental datasets, hidden size and batch size is set to 64. Following the common practice in sequential recommendations, we leave the interactions happening at the latest time as the test dataset and the interactions at the second latest time as the validation dataset.

Table 2: Overall performance. Bold scores represent the highest results of all methods. Underlined scores stand for the second-highest results. Our model achieves the state-of-the-art result among all baseline models. * means the improvement is significant at $p < 0.05$.

Dataset	Metric	BPR	GRU4Rec	Caser	NARM	SASRec	TiSASRec	SLRS+	Chorus	AHMP	KDA	SDIL	Improv.
Beauty	HR@5	0.3317	0.3202	0.3210	0.3334	0.4004	0.3872	0.4339	0.4536	0.4566	0.4860	**0.4926***	1.36%
	HR@10	0.4355	0.4311	0.4345	0.4462	0.5074	0.4980	0.5337	0.5698	0.5519	0.5997	**0.6128***	2.18%
	HR@20	0.5505	0.5693	0.5757	0.5823	0.6268	0.6179	0.6361	0.6838	0.6599	0.7144	**0.7323***	2.51%
	NDCG@5	0.2361	0.2271	0.2246	0.2348	0.2923	0.2904	0.3319	0.3386	0.3496	0.3648	**0.3698***	1.37%
	NDCG@10	0.2697	0.2628	0.2612	0.2712	0.3268	0.3181	0.3642	0.3762	0.3803	0.4016	**0.4088***	1.79%
	NDCG@20	0.2987	0.2976	0.2967	0.3055	0.3569	0.3483	0.3900	0.4050	0.4076	0.4306	**0.4390***	1.95%
	MRR	0.2363	0.2271	0.2246	0.2366	0.2923	0.2904	0.3319	0.3386	0.3421	0.3549	**0.3610***	1.72%
Cellphone	HR@5	0.3387	0.3015	0.3937	0.4168	0.4586	0.4520	0.4696	0.4697	0.5045	0.5497	**0.5538***	0.75%
	HR@10	0.4528	0.4301	0.5309	0.5509	0.5810	0.5767	0.5641	0.5929	0.6132	0.6745	**0.6792***	0.70%
	HR@20	0.5852	0.5918	0.6810	0.6974	0.7067	0.7022	0.6637	0.7152	0.7284	0.7923	**0.8028***	1.33%
	NDCG@5	0.2430	0.2085	0.2800	0.2995	0.3412	0.3344	0.3634	0.3530	0.3852	0.4119	**0.4188***	1.69%
	NDCG@10	0.2798	0.2498	0.3243	0.3429	0.3809	0.3748	0.3939	0.3929	0.4204	0.4523	**0.4595***	1.59%
	NDCG@20	0.3131	0.2905	0.3622	0.3799	0.4126	0.4065	0.4191	0.4238	0.4495	0.4821	**0.4908***	1.80%
	MRR	0.2453	0.2271	0.2246	0.2969	0.2923	0.2904	0.3319	0.3386	0.3747	0.3666	**0.4049***	10.45%
Toys	HR@5	0.2897	0.2902	0.2898	0.3173	0.3687	0.3475	0.4368	0.4124	0.4603	0.4805	**0.4953***	3.08%
	HR@10	0.3897	0.4060	0.4103	0.4336	0.4767	0.4608	0.5345	0.5203	0.5587	0.5882	**0.6069***	3.18%
	HR@20	0.5061	0.5546	0.5590	0.5777	0.6018	0.6003	0.6440	0.6443	0.6621	0.7019	**0.7248***	3.26%
	NDCG@5	0.2068	0.1974	0.1947	0.2206	0.3023	0.2535	0.3490	0.3132	0.3600	0.3660	**0.3797***	3.74%
	NDCG@10	0.2390	0.2348	0.2336	0.2581	0.3140	0.2901	0.3804	0.3480	0.3918	0.4007	**0.4157***	3.74%
	NDCG@20	0.2683	0.2721	0.2710	0.2944	0.3339	0.3253	0.4081	0.3793	0.4179	0.4294	**0.4454***	3.73%
	MRR	0.2116	0.2271	0.2246	0.2244	0.2923	0.2904	0.3319	0.3386	0.3547	0.3666	**0.3713***	1.28%

5.1.2 Baselines for Comparisons. To evaluate the performance of our model TPNE, we select 10 representative and/or state-of-the-art recommendation models as baselines. They are classified into the following three groups: **(1) Non-sequential models:** *BPR* [20]: This model characterizes the pairwise interactions via a matrix factorization model and optimizes through a pair-wise Bayesian Personalized Ranking loss. **(2) Non-time-sensitive sequential models:** *GRU4Rec* [10]: This model uses the GRU to model the user interaction sequence and gives the final recommendation. *NARM* [16]: It is a classic long- and short-term user interest modeling, which designs two RNN-based global encoders and a local encoder to model the user's long and short-term interest respectively. *Caser* [36]: This model embeds items in user interaction history as image form, using convolutional filters for the recommendation. *SASRec* [14]: This model leverages users' longer-term semantics as well as their recent actions simultaneously for accurate next-item recommendations. **(3) Time-sensitive sequential models:** *TiSASRec* [17]: This model leverages the timestamp in the user-item interaction, exploring the different time intervals' influence in the next item prediction. *SLRS+* [26]: SLRS combines Hawkes process and MF into one framework, which aims at modeling the user repeat consumption in the sequential recommendation. Since the Amazon dataset removes the repeat consumption in the test set, SLRS+ uses the Hawkes process to model the relations including also view and also buy behaviors. *AMHP* [31]: This model introduces the repeat category consumption into sequential recommendation and adopts the self-attention architecture as the main model. *Chorus* [27]: This model is a state-of-the-art method with

item relations and temporal evolution. *KDA* [25]: KDA devises relational intensity and frequency-domain embeddings to adaptively determine the importance of historical interactions.

5.1.3 Implementation Details and Parameter Settings. To ensure fair comparisons, we follow the ReChorus' baseline implementation[1]. Specifically, for both our model and all baseline models, we set the embedding size and batch size to 64. The number of training epochs for all models is set to 150 and an early-stop strategy is used. When a model's performance on the validation set decreases for 10 consecutive rounds, the training will stop.

For fair comparison, we carefully tuned the model parameters of all the baselines on the validation set. Then we choose the model parameters that achieve the best performance on the validation set to compare. In Caser, the number of horizon convolution kernels is set as 64, the number of vertical convolution kernels is set as 32 and the union window size is set as 5. In GRU4Rec, the dimension of the hidden layer is set as 64. In SARS and TiSARS, the number of heads is set as 1. In SLRS+ and Chorus, the learning rate is set as 5e-4. In AHMP, both the head number and layer number are set as 1. The learning rate is set as 1e-4. In KDA, the number of heads is set as 4, and the learning rate is set as 1e-3. Besides, the learning of all the models is carried out five times to report the average results.

For our SDIL model, we set the number of heads in the multi-head attention model of SIM to 4 and the number of layers of both DIM and SIM to 2 by using a grid search from 1 to 8 with a step of 1. We set the embedding dimension of all the attributes including price, brand, and category to 64. We pre-trained the item embedding

---

[1]https://github.com/THUwangcy/ReChorus

with a learning rate of 5e-4 and the main model is optimized by Adam optimizer with a learning rate of 1e-4.

5.1.4 **Evaluation Metrics**. We use three commonly used metrics including *Normalized Discounted Cumulative Gain @K* (NDCG@K), *Hit Ratio @K* (HR@K) and *Mean Reciprocal Rank* (MRR) [14, 27] to evaluate the performance of all compared methods. For all the baseline models, we generate the ranking list of items for each testing interaction. Both of them are applied with *K* chosen from {5, 10, 20}. In order to accelerate the evaluation, following the work [27], we evaluate the ranking results with 99 randomly selected negative items. Meanwhile, we follow the setting in[28], using the paired t-test with p<0.05 for the significance test.

5.2 Performance Comparison with Baselines

Table 2 reports the comparison results between our method and 10 different baseline methods on three datasets. The best results and the second-best results across all methods are in bold and underlined respectively. We have the following observations:

(1) First, we can find that all the non-time-sensitive sequential methods such as GRU4Rec [10, 11], Caser [36] surpass the non-sequential method BPR, which indicates the effectiveness of modeling sequential information. Meanwhile, NARM utilizes both global and local encoders to model the long- and short-term interest of users and thus performs better than GRU4Rec and Caser. Moreover, the self attention-based SRS, i.e., SASRec [14], consistently outperforms previous traditional deep learning methods including CNN-based Caser and RNN-based GRU4Rec, proving the effectiveness of using the attention mechanism to encode sequence data.

(2) Second, more advanced time-sensitive sequential models, such as TiSASRec [17], generally inherit the attention-based encoder while introducing extra-temporal signals to further improve the recommendation performance. Specifically, TiSASRec explicitly models both the absolute positions of items as well as time intervals between them in a sequence, and thus it outperforms SARS w.r.t most evaluation metrics on most datasets. SLRS+ [26] introduces both knowledge-graph information and the multi-excitation of items to further enhance the recommendation performance. Chorus [27] further models the different situations of the temporal effect of user-item interactions on item embedding level and thus outperforms SLRS+. However, due to the limited representation ability of its utilized base model, Chorus shows the over-fitting issue. AHMP [31] models the user temporal category-level consumption excitation via self-attention models, achieving the third-best results on most datasets. KDA [25] designs the virtual relations between items to enhance the relation learning between items, which helps KDA obtain a significant improvement over other baselines.

(3) Third, our proposed SDIL model consistently outperforms all baselines in all datasets. The average improvement compared with the best-performing baseline is 10.45% w.r.t MRR in Cellphone dataset. The reason is obvious. Different from most of the existing SRS methods, which ignore modeling either users' relatively stable static interest, or the negative excitation of historical interactions on user's next choice, our proposed SDIL model has been carefully designed to well address these issues. To be specific, SDIL not only well models users' static interest in items via carefully capturing users' interest on item attribute level, but also comprehensively

models users' dynamic interest via the well-designed TPNE module. More importantly, TPNE is able to model both positive and negative excitation of users' historical interactions on their next choice, leading to significant performance improvement.

5.3 Ablation Study

To analyze the rationality and effectiveness of our designed components in the SDIL model, we conduct the ablation study with three variants. **SDIL-1**: Only uses static interest module to predict the next item; **SDIL-2**: Only uses dynamic interest module to predict the next item; **SDIL-3**: Uses both dynamic and static interest module to predict the next item without modeling excitation. Figure 3 presents the performance of these three variants and the full model *SDIL* on the three experimental datasets. First, compare *SDIL-1* and SDIL, we find the performance of *SDIL-1* significantly decreases when dropping the dynamic interest modeling module TPNE from *SDIL*. This proves the extremely importance of our designed TPNE module. Second, the comparison between *SDIL-2* and *SDIL* demonstrates that static interest modeling is also vital for recommendation performance. The performance clearly decreases without it. Third, the comparison between *SDIL-3* and *SDIL* demonstrates the importance of modeling historical interactions' excitation on users' next choice. In summary, the dropping of any of our designed components would lead to a clear reduction in recommendation performance, this effectively verifies the effectiveness and rationality of our design.

5.4 The Effectiveness of Modeling Negative Excitation

To verify the rationality of incorporating negative excitation and the effectiveness of our proposed TPNE module, we keep the SDIL framework while changing its TPNE module to TPE to obtain another variant SDIL-TPE. TPE models users' dynamic interest by only considering temporal *positive* excitation only. As Table 3 shows, SDIL consistently surpasses SDIL-TPE, indicating that the comprehensive modeling of both positive and negative excitation is more effective compared with the modeling of only positive excitation.

5.5 Parameter Sensitivity Test & Case Study

We conduct the parameter sensitivity test to see how the key model parameters will affect the performance of our SDIL model. The result is shown in Figure 5. In addition, we conduct case studies to show some insights on how our proposed model can work better in a straightforward way. Due to the limited space, the details are provided in Figure 6 and Figure 7. In this section, we will discuss the parameter setting's effect on the model performance. The results are presented in Figure 5. The embedding size of item representation does affect the performance of SRS models. We can see that the performance of the model increases consistently as the embedding size increases.

Besides, we also conduct experiments about the different multi-head numbers and transformer layers. The head number of TPNE layers is searched from {1,2,4,6} and the Transfomer layer number is searched from {0,1,2,3,4}. We can find that the performance increases as the head number increases, while when the layer or head number is too large, the model begins to over-fit.

(a) Beauty Dataset.
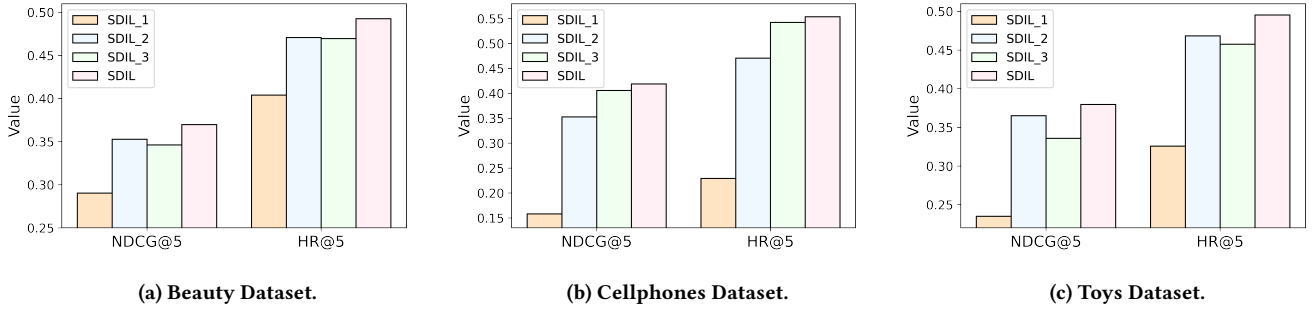
(b) Cellphones Dataset.

(c) Toys Dataset.

Figure 3: Ablation Study on the Model Performance. (HR@5 and NDCG@5) on different datasets.

Table 3: Performance comparison between SDIL-TPE and SDIL. * means the improvement is significant at $p < 0.05$.

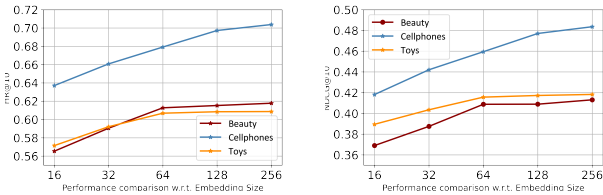| Dataset	Metrics	SDIL-TPE	SDIL
Beauty	HR@5	0.4825	0.4926*
	HR@10	0.6054	0.6128*
	NDCG@5	0.3487	0.3698*
	NDCG@10	0.4014	0.4088*
	MRR	0.3534	0.3602*
Cellphones	HR@5	0.5521	0.5538*
	HR@10	0.6772	0.6792*
	NDCG@5	0.4102	0.4188*
	NDCG@10	0.4422	0.4595*
	MRR	0.4038	0.4049*
Toys	HR@5	0.4871	0.4953*
	HR@10	0.5979	0.6069*
	NDCG@5	0.3741	0.3797*
	NDCG@10	0.3670	0.4157*
	MRR	0.3670	0.3713*

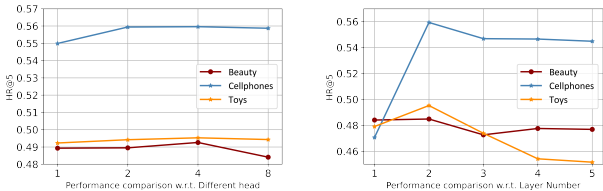Figure 4: Embedding size setting's effect on the model performance. (HR@5 and NDCG@10).

Figure 5: Different transformer layers setting's effect on the model performance. (HR@10 and NDCG@10).

To get a better understanding of the effectiveness of our TPNE in a specific context, we randomly select a user sequence and list the recommendation results of TPNE and its variant TPE. In Figure 6, the user's interaction history demonstrates that the user first purchased the Samsung S4 phone and then purchased the corresponding accessories, including the wallet case holder, car phone holder, and screen protector. For the next-item prediction task, the ground-truth item is a blue-tooth earpiece. For the TPE using only positive incentive learning, although it can capture the positive excitation between the phone and the earpiece, it also introduces the positive excitation with the screen protector. However, the user has already purchased the screen protector in the previous moment, so it will lead to homogeneous recommendations, resulting in inaccurate recommendation results. As for TPNE, our model considers the negative excitation of similar items in the short term, in this case, for the similar screen protector items in timestamp t, the model will come out with a negative mutual exclusion signal. Thus, the homogeneous recommendation can be effectively alleviated, and better recommendation performance can be obtained. It verifies the importance of considering both positive and negative excitation learning. Moreover, to prove the superiority of SDIL in combining both dynamic and static interest contexts. We listed the ranking results of DIL, SIL, and SDIL. As Figure 7 illustrated, we random sample one user interaction item history. We can also find our SDIL can well capture the user intent in the next-item prediction task.

6 CONCLUSION

In this paper, we propose a novel Static-Dynamic Interest Learning (SDIL) framework to comprehensively model users' static interest and dynamic interest in items for generating accurate sequential recommendations. More importantly, to comprehensively model users' dynamic interest, we devise a novel Temporal Positive and Negative Excitation modeling (TPNE) module. TPNE can not only well capture both the positive and negative excitation of each user's historical interactions on his/her next choice of items, but also is able to specify the excitation strength for each historical interaction according to its happening time. Our extensive experimental results on four real-world datasets validate the superiority of our proposed model over state-of-the-art methods. In the future, we will explore how to better model users' static interest to further enhance the performance of our model.
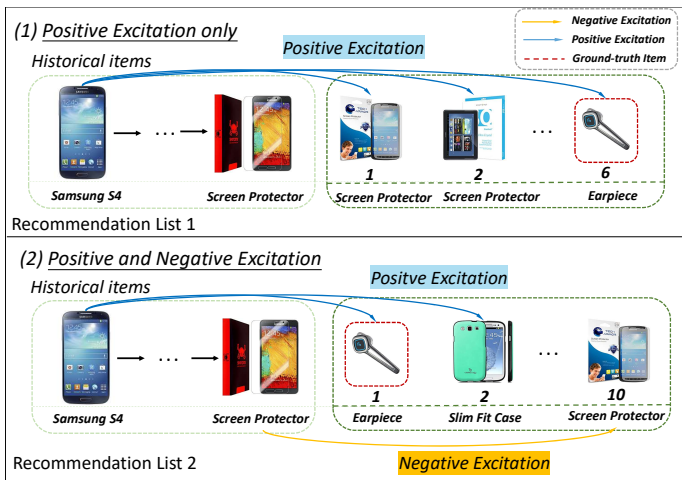
Figure 6: Illustration of the ranking results of TPE and TPNE. The item highlighted in the red boxes is the ground-truth item.

Figure 7: Illustration of the ranking results of SIM, DIM and DSIM. The item highlighted in the red boxes is the ground-truth item.

REFERENCES

[1] Renqin Cai, Xueying Bai, Zhenrui Wang, Yuling Shi, Parikshit Sondhi, and Hongning Wang. 2018. Modeling Sequential Online Interactive Behaviors with Temporal Point Process. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, October 22-26, 2018*. ACM, 873–882.

[2] Yongjun Chen, Zhiwei Liu, Jia Li, Julian J. McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*. ACM, 2172–2182.

[3] Angelos Dassios and Hongbiao Zhao. 2013. Exact simulation of Hawkes process with exponentially decaying intensity. *Electronic Communications in Probability* 18 (2013), 1–13.

[4] Seyda Ertekin, Cynthia Rudin, and Tyler H. McCormick. 2013. Predicting Power Failures with Reactive Point Processes. In *Late-Breaking Developments in the Field of Artificial Intelligence, Bellevue, Washington, USA, July 14-18, 2013*, Vol. WS-13-17. AAAI.

[5] Ziwei Fan, Zhiwei Liu, Jiawei Zhang, Yun Xiong, Lei Zheng, and Philip S. Yu. 2021. Continuous-Time Sequential Recommendation with Temporal Graph Collaborative Transformer. In *The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 433–442.

[6] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[7] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *IEEE 16th International Conference on Data Mining, December 12-15, 2016, Barcelona, Spain*. IEEE Computer Society, 191–200.

[8] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web, 2016, Montreal, Canada, April 11 - 15, 2016*. ACM, 507–517.

[9] Ruining He and Julian J. McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*. AAAI Press, 144–150.

[10] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent Neural Networks with Top-k Gains for Session-based Recommendations. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, Torino, Italy, October 22-26, 2018*. ACM, 843–852.

[11] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

[12] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying Personalized Recommendation with User-session Context. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, August 19-25, 2017*. 1858–1864.

[13] Dietmar Jannach and Malte Ludewig. 2017. When Recurrent Neural Networks meet the Neighborhood for Session-Based Recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, August 27-31, 2017*, Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin (Eds.). ACM, 306–310.

[14] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *IEEE International Conference on Data Mining, Singapore, November 17-20, 2018*. IEEE Computer Society, 197–206.

[15] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*. ACM, 447–456.

[16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, Singapore, November 06 - 10, 2017*. ACM, 1419–1428.

[17] Jiacheng Li, Yujie Wang, and Julian J. McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. ACM, 322–330.

[18] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory Augmented Graph Neural Networks for Sequential Recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, February 7-12, 2020*. AAAI Press, 5045–5052.

[19] Julian J. McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*. ACM, 43–52.

[20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR:Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*. AUAI Press, 452–461.

[21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web , Raleigh, North Carolina, USA, April 26-30, 2010*. ACM, 811–820.

[22] Frederic Paik Schoenberg, David R Brillinger, and PM Guttorp. 2002. Point processes, spatial-temporal. *Encyclopedia of environmetrics* 3 (2002), 1573–1577.

[23] Guy Shani, Ronen I. Brafman, and David Heckerman. 2002. An MDP-based Recommender System. In *Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence, Edmonton, Alberta, Canada, August 1-4, 2002*. Morgan Kaufmann, 453–460.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, Vol. 30. Curran Associates, Inc.

[25] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2021. Toward Dynamic User Intention: Temporal Evolutionary Effects of Item Relations in Sequential Recommendation. *ACM Transactions on Information Systems* 39, 2 (2021), 16:1–16:33.

[26] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference, San Francisco, CA, USA, May 13-17, 2019*. ACM, 1977–1987.

[27] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make It a Chorus: Knowledge- and Time-aware Item Modeling for Sequential Recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, Virtual Event, China, July 25-30, 2020*. ACM, 109–118.

[28] Meirui Wang, Pengjie Ren, Lei Mei, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. A Collaborative Session-based Recommendation Approach with Parallel Memory Modules. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, July 21-25, 2019*. ACM, 345–354.

[29] Shoujin Wang, Liang Hu, Longbing Cao, Xiaoshui Huang, Defu Lian, and Wei Liu. 2018. Attention-Based Transactional Context Embedding for Next-Item Recommendation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. AAAI Press, 2532–2539.

[30] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, August 10-16, 2019*. 6332–6338.

[31] Xilin Wen, Jianfang Wang, Xu Yang, and Qiuling Zhang. 2021. A Sequential Recommendation of Hawkes Process Fused with Attention Mechanism. In *The 23rd IEEE International Conference on High Performance Computing and Communications, Haikou, Hainan, China, December 20-22, 2021*. IEEE, 301–308.

[32] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2020. Contrastive learning for sequential recommendation. *arXiv preprint arXiv:2010.14395* (2020).

[33] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff G. Schneider, and Jaime G. Carbonell. 2010. Temporal Collaborative Filtering with Bayesian Probabilistic Tensor Factorization. In *Proceedings of the SIAM International Conference on Data Mining, April 29 - May 1, 2010, Columbus, Ohio, USA*. SIAM, 211–222.

[34] Ghim-Eng Yap, Xiaoli Li, and Philip S. Yu. 2012. Effective Next-Items Recommendation via Personalized Sequential Pattern Mining. In *Database Systems for Advanced Applications - 17th International Conference, 2012, Busan, South Korea, April 15-19*, Vol. 7239. Springer, 48–64.

[35] Dongjin Yu, Jianjiang Chen, Dongjing Wang, Yueshen Xu, Zhengzhe Xiang, and Shuiguang Deng. 2022. DSIM: dynamic and static interest mining for sequential recommendation. *Knowledge and Information Systems* 64, 8 (2022), 2267–2288.

[36] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. 2019. A Simple Convolutional Generative Network for Next Item Recommendation. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, Melbourne, VIC, Australia, February 11-15, 2019*. ACM, 582–590.

[37] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. 2019. Next item recommendation with self-attentive metric learning. In *Thirty-Third AAAI Conference on Artificial Intelligence*, Vol. 9.

[38] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, August 10-16, 2019*. 4320–4326.

[39] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 1893–1902.

[40] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to Do Next: Modeling User Behaviors by Time-LSTM. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, August 19-25, 2017*. 3602–3608.