Abstract—At the age of big data, recommender systems have shown remarkable success as a key means of information filtering in our daily life. Recent years have witnessed the technical development of recommender systems, from perception learning to cognition reasoning which intuitively build the task of recommendation as the procedure of logical reasoning. However, the logical statement in reasoning implicitly admits irrelevance of ordering, even does not consider time information which plays an important role in recommendation. Furthermore, recommendation model incorporated with temporal context would tend to be self-attentive, i.e., automatically focus more (less) on the relevance (irrelevance), respectively.

In this paper, we propose a Time-aware Self-Attention with Neural Collaborative Reasoning (TiSANCR) based recommendation model, which integrates temporal patterns and self-attention mechanism into reasoning-based recommendation. Special, temporal patterns represented by relative time provide context and auxiliary information to characterize the user’s preference in recommendation, while self-attention is leveraged to distill informative patterns and suppress irrelevances. Extensive experiments on benchmark datasets demonstrate that the proposed TiSANCR achieves significant improvement and consistently outperforms the state-of-the-art recommendation methods.

Index Terms—Recommendation Systems, Time-aware, Self-Attention, Logic Reasoning

I. INTRODUCTION

Over the past decades, the explosive growth of data has produced large number of redundant information, leading to information overload [4]. Recommender system (RS) has become prevalent technique of information filtering for alleviating this issue, and learns user’s previous interactions with items (e.g., ratings, clicks, add-to-favorite, dislike and purchase). Collaborative filtering is a dominant methodology which relies on information provided by users who share same preference, i.e., leveraging the wisdom of crowd. However, trivially multiplying of latent embeddings linearly, the inner product may not be sufficient to capture the complex structure of user-item interactions well.

Deep neural networks (DNN) have been further extended to CF methods, deploying complex structures of neural networks to improve performance of RS, including AutoRec [15], Neural Collaborative Filtering (NCF) [6], CDAE [18], CDL [17], DeepCoNN [21], to name a few. However, it is controversial whether complex architecture of DNN is superior to simple matching function [2], [3]. A recent line of researches including NCR [1] and LINN [16], focus on reasoning-based models, in which the procedures of recommendation exhibit certain expressions of propositional logic.

Reasoning-based recommendations achieve significant improvement on ranking performance [1], [16], however, propositional logic modeled by neural modules have not taken time information of interactions into consideration. On one hand, logic expressions including conjunction and disjunction operations, neglect the ordering of events. For example, the conjunction $a \land b$ which represents that one user bought item $a$ and $b$, implicitly admits irrelevance of ordering due to the commutative law of conjunction, i.e., $a \land b \iff b \land a$. However, the ordering of purchase list should have great impacts on the results of recommendation. For instance, if a user purchased T-shirt and then bought heels, one would expect the user to purchase socks to make collocation to heels. In contrast, leisure shorts would be recommended to the user if he/she purchased heels and bought T-shirt three months later.

On the other hand, temporal recommendation [10], [20] typically model the temporal dynamics of user’s preference, which characterizes user time-sensitive behaviors. However, exploiting temporal information in recommendation, to some extent, does not provides highlighted correlations of interactions in context. As exampled in Fig. 1, in the upper dashed rectangle, the mother is recommended to purchase 1st stage baby formulas with high probability. On the contrary, as shown in the lower rectangle, 2nd stage baby formulas should be more reasonable since 2nd stage baby formulas is suitable for 6 month old baby. Therefore, the relative time 6 months provides significant information in context to the recommendation of 2nd stage baby formulas. Under this condition, the model of RS would tend to be self-attentive on temporal information.

In order to enhance the influence of temporal information on recommendation, the recent advance in DNN model - the attention mechanism, is commonly employed in recommendation system [8], [11]. The self-attention mechanism provides guidance to model to distill informative patterns and suppress
irrelevance. In reasoning-based recommendation, the logical value of material implication \((a \land b) \rightarrow c\) which indicates whether the item \(c\) is recommended or not when the interactions \(a \land b\) were observed, could be heavily dependent on the parts of observations \(a\) and/or \(b\). Hence, it is an open question whether integrating temporal information and self-attention into reasoning-based recommendation system contributes to improving performance?

In this paper, we answer this question to fulfill this research gap by proposing TiSANCR (Time-aware Self-Attention with Neural Collaborative Reasoning) based recommendation system, which bridges self-attentive temporal information to the NCR framework. Specifically, in order to capture temporal dynamics of logic reasoning, we extract relative time of each interaction as context information. In order to enhance the influence of temporal information on recommendation, we build self-attention module to adaptively capture weights of previous interactions in terms of timestamp.

The remainder of this paper is organized as follows. The proposed TiSANCR model is presented in Section II. In Section III, we conduct experiments to compare TiSANCR with state-of-the-art baselines over public datasets. At last, we provide the conclusion of this study in Section IV.

II. PROPOSED METHOD

A. Problem Formulation

Let \(U\), \(V\) and \(T\) denote the sets of the users, items and timestamps respectively, \(|U|\), \(|V|\) and \(|T|\) denote the total number of users, items and timestamps respectively. For each user \(u \in U\), his/her historical events of interaction are organized chronologically as \(E_u = \{e_{u1,t1}, e_{u2,t2}, \cdots, e_{ur,tr}\}\), where \(r\) is the total number of items in the historical interactions, \(e_{ui,ti}\) means that the user \(u\) interacted with item \(vi\) at time \(ti\), where \(v1 \in V\) and \(t1 \leq t2 \leq \cdots \leq tr\) holds.

The task of temporal recommendation is to decide whether the next item \(vt\) at timestamp \(tr\) is recommended to the user \(u\), according to his/her historical behaviors of interaction \(E_u\). Hence, together with temporal information, recommendation based on historical interactions can be formally defined to:

\[
\begin{align*}
(e_{u1,t1} & \land e_{u2,t2} \land \cdots \land e_{ur,tr}) \rightarrow e_{ur,tx}, \\
\text{which could be further rewritten as follow according to De Morgan’s Law}
\end{align*}
\]

\[
\begin{align*}
\neg e_{u1,t1} \lor \neg e_{u2,t2} \lor \cdots \lor \neg e_{ur,tr} & \lor e_{ur,tx}.
\end{align*}
\]

B. Extension to Explicit Feedback

The procedure of reasoning-based recommendation in Eq. (1) considers the implicit feedback of user, i.e., each historical event \(e_{ui,ti}\) only reveals the interaction between user \(u\) and item \(vi\), but no information about whether \(u\) likes or dislikes \(vi\). When extending it to reasoning-based recommendation, inspired by NCR [1], we define \(e_{ui,t}\) and \(\neg e_{ui,t}\) as positive feedback and negative feedback, respectively. Suppose that the historical events of interaction of user \(u\) \(E_u = \{e_{u1,t1}, e_{u2,t2}, \cdots, e_{ur,tr}\}\) describes the attitude of user \(u\) towards interacted items \(vi\) at timestamp \(ti\). Whether or not to recommend the candidate item \(v2\) at time \(tx\) could be expressed as the following logical statement:

\[
(e_{u1,t1} \land e_{u2,t2} \land \cdots \land \neg e_{ur,tr}) \rightarrow e_{ur,tx}.
\]

Note that the negative feedbacks could be on more items.

C. Network Architecture

The overall architecture of our model is shown in Fig. 2. First of all, given a user \(u\), his/her interacted items \(v1\) at timestamp \(t1\) \((i = 1, \cdots, r)\) and candidate item \(v2\) at current timestamp \(tr\), the user \(u\), items \(\{v1, \cdots, vr, v2\}\) and timestamps \(\{t1, \cdots, tr\}\) are embedded to dense vectors in a differential space respectively. To highlight the impact of temporal information on reasoning-based recommendation, we consider the relative time to represent the temporal information, which are then fed into the self-attention layer to capture the importance of temporal patterns.

On the other hand, the embedded vector of user and item are concatenated, which are further fed into an encode layer to encode the historical events. The encoded historical events and the corresponding historical timestamps outputted by self-attention layer are simply added to represent the interactive events with temporal information, which are then fed into LNN layer, together with the concatenation of the embedded vector of user and that of candidate item. At last, the similarity between the output vector of LNN layer and a constant true vector can be calculated for training the parameters of the model. In the rest of this subsection, we introduce each part of TiSANCR model in details.

Embedding Layer: Due to limited representation capacity of original user IDs, item IDs and timestamps, we employ three embedding layers to embed the users, items IDs, and the timestamp respectively, into three continuous low-dimensional spaces. Each embedding layer is implemented by fully connected network to map the sparse inputs to dense vectors. Formally, given the sparse input features of user \(\tilde{u}\), item \(\tilde{v}\) and timestamp \(\tilde{t}\), we can obtain their dense embeddings \(\tilde{u} \in \mathbb{R}^d\), \(\tilde{v} \in \mathbb{R}^d\) and \(\tilde{t} \in \mathbb{R}^d\) via

\[
\tilde{u} = E_u^T \tilde{u}, \quad \tilde{v} = E_v^T \tilde{v}, \quad \tilde{t} = E_t^T \tilde{t}
\]

where \(E_u \in \mathbb{R}^{|U| \times d}\), \(E_v \in \mathbb{R}^{|V| \times d}\) and \(E_t \in \mathbb{R}^{|T| \times d}\) are the embedding matrices for user features, item features and temporal features respectively, and \(d\) is the dimension of the latent embedding space.

Self-attention Layer: To highlight the impact of temporal information, we consider relative time to capture temporal information. Formally, given the embedding of timestamps \(\tilde{t}_i (i = 1, \cdots, r)\) in historical interactions and current timestamp \(\tilde{t}_r\), the relative time is calculated by \(\tilde{t}_i = \tilde{t}_r - \tilde{t}_i\). Note that we overuse \(\tilde{t}_i\) here, since the relative time can easily recover the timestamp according to current timestamp \(\tilde{t}_r\).

Given a sequence of relative time in historical interactions \(\{\ell_1, \ell_2, \cdots, \ell_r\}\), the self-attention mechanism computes a corresponding sequence of “highlighted” temporal information \(\{\ell_1, \ell_2, \cdots, \ell_r\}\). Each output element in the highlighted
sequence $\ell_i$, is computed as a weighted sum of linearly transformed embeddings of centralized relative times, formally,

$$\ell_i = \sum_{j=1}^{r} w_{ij} \left( W^V t_j \right),$$

where $W^V \in \mathbb{R}^{d \times d}$ is the trainable transformation matrix for value. Each weight coefficient $w_{ij}$ is computed using a softmax function:

$$w_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{k=1}^{r} \exp(\alpha_{ik})},$$

where $\alpha_{ij}$ is the contextual relevance of relative timestamps $t_i$ and $t_j$, and the relevance is measured via the dot product:

$$\alpha_{ij} = \frac{(W^Q t_i) (W^K t_j)^T}{\sqrt{d}},$$

where $W^Q \in \mathbb{R}^{d \times d}$ and $W^K \in \mathbb{R}^{d \times d}$ are the trainable projection matrices to distill useful time information for propagation of logical reasoning. The denominator $\sqrt{d}$ is the limit coefficient which prevents the value of the inner product above from being too large, especially when the dimension is high.

**Encoder Layer:** To model the non-linear interactions between user and item, MLP is commonly to fuse the dense embeddings of user and item. We adopt a simple neural network to encode the non-linearity of interaction between user embedding $u$ and item embedding $v$ into event vector $e_u^i$, which we regard as non-linear fusion. The neural network takes the concatenation of user and item embeddings as input, and contains two linear transformations with a ReLU activation in between, formally,

$$e_u^i = W_2 \phi(W_1 (u \oplus v) + b_1) + b_2$$

where $u, v \in \mathbb{R}^d$ are user and item latent embedding vectors in $d$-dimensional space; $\oplus$ is concatenation operation between user and item embeddings; $W_1 \in \mathbb{R}^{2d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$ are weight matrices and $b_1, b_2 \in \mathbb{R}^d$ are bias terms; $e_u^i \in \mathbb{R}^d$ represents the encoded event embedding, $\phi(\cdot) = \max(0, x)$ is the rectified linear unit (ReLU) activation function.

Furthermore, in order to fuse temporal patterns outputted by self-attention layer into user-item interactions, we encode each temporal event vector $e_u^{i_1, t_1}$ by simply adding self-attentive time information $\ell_i$ accordingly to event vector $e_u^i$, i.e., $e_u^{i_1, t_1} = e_u^i + \ell_i$, $(i = 1, \ldots, r)$. It is worth mentioning that the temporal event vector $e_u^{i_1, t_1}$ degenerates to the event vector $e_u^i$. Since that the relative time of current timestamp is zero, the self-attentive time information $\ell_x$ is the arithmetic average of relative time embeddings $t_i$, i.e., $\ell_x = \frac{1}{r} \sum_{i=1}^{r} W^V t_i$, leading to no attention of $\ell_x$ on previous temporal information. Hence, it is non-trivial to fuse event embedding and highlighted temporal embedding by applying addition between them.

**LNN Layer:** Based upon the fusion of event embeddings and highlighted temporal embeddings, the next step is to utilize logical neural network to predict the logical expression in Eq.(2), which is represented as the logical aggregation of temporal event embeddings:

$$(-e_u^{i_1, t_1} \lor -e_u^{i_2, t_2} \lor \cdots \lor -e_u^{i_r, t_r}) \lor e_u^{x, t_x}.$$

In order to carry out the prediction of logical expression, NCR [1] employs LNN layer to perform logical reasoning based on the embeddings of interaction events. We follow this pipeline, but the difference is that the prediction of logical reasoning is based on the temporal event vectors as shown in Eq.(9), which fuses not only interaction events but also highlighted temporal information.

According to the logical expression in Eq.(9), we firstly calculate the negated temporal event $-e_u^{i_1, t_1}$ by feeding each input temporal event $e_u^{i_1, t_1}$ into NOT layer:

$$-e_u^{i, t_i} = \text{NOT}(e_u^{i, t_i}), \quad \forall i \in \{1, 2, \ldots, r\}$$

Fig. 2. The overall architecture of TISANCR.
After that, along with above negated temporal events \( -e_{v_i,t_i}^x \), the candidate temporal event \( e_{v_i,t_i}^{x,t} \) is fed into the OR layer, of which the output generates the final embedding of the logical expression in Eq.(9) as follow

\[
\text{Exp} = \text{OR}(e_{v_i,t_i}^{x,t}, -e_{v_i,t_i}^{y,t}, ..., -e_{v_i,t_i}^{r,t}, e_{v_i,t_i}^{s,t_z}) \tag{11}
\]

where \( \text{Exp} \) is the vector representation of the logical expression, and the operator OR is the logical disjunction of all input temporal event vectors. This can be implemented by recurrently calling the basic binary model OR(\( \cdot \), \( \cdot \)) as shown at line 5 in following procedure:

**Procedure:** LNN\((e_{v_i,t_i}^{x,t}, e_{v_i,t_i}^{y,t}, ..., e_{v_i,t_i}^{r,t}, e_{v_i,t_i}^{s,t_z})\)

1. Input: \( e_{v_i,t_i}^{x,t}, e_{v_i,t_i}^{y,t}, ..., e_{v_i,t_i}^{r,t}, e_{v_i,t_i}^{s,t_z} \)
2. Initialization: \( \text{Exp} = e_{v_i,t_i}^{x,t}, I = \{1, \ldots, r\} \)
3. While \( I \) is not empty:
   4. Randomly draw \( i \) from \( I \) without replacement.
   5. \( \text{Exp} \leftarrow \text{OR}(\text{Exp}, \text{NOT}(e_{v_i,t_i}^{i,t})) \)
6. While end
7. Return: \( \text{Exp} \)

### D. Loss Function

After obtaining the vector representation of logical expression in Eq.(9), the model should determine whether \( \text{Exp} \) represents true or false. To this end, the vector representation is examined in the logical space by computing the distance during model training and inference. We employ the cosine similarity to measure this distance:

\[
\text{Sim}(\text{Exp}, T) = \frac{(\text{Exp}, T)}{||\text{Exp}|| ||T||}. \tag{12}
\]

We employ the pair-wise learning algorithm [14] for model training. For a user \( u \), according to his/her interactions, the set of items \( \mathcal{V} \) can be divided to two disjoint set \( \mathcal{V}_u^+ \) and \( \mathcal{V}_u^- \), where \( \mathcal{V}_u^+ \) represents those of that the user \( u \) interacted with, while \( \mathcal{V}_u^- \) represents those of that the user \( u \) did not interact with. \( \mathcal{V}_u^+ \cap \mathcal{V}_u^- = \emptyset \) and \( \mathcal{V}_u^+ \cup \mathcal{V}_u^- = \mathcal{V} \) hold. Let the pairs of items and timestamps \( \{(v_1, t_1), (v_2, t_2), ..., (v_r, t_r)\} \) be the historical interactions of user \( u \). We sample item \( v_x \in \mathcal{V}_u^+ \) with timestamp \( t_x \) to construct positive sample that user \( u \) interacted with, and item \( v_y \in \mathcal{V}_u^- \) with timestamp \( t_y \) to construct negative sample that user \( u \) did not interact with.

The logical expressions of positive sample and counter sample recommendation could be modelled as

\[
\begin{align*}
E_{u,x}^+ &= (-e_{u,t_x}^{1,t_1} \lor -e_{u,t_x}^{2,t_2} \lor \ldots \lor -e_{u,t_x}^{r,t_r}) \lor e_{u,t_x}^{s,t_z} \\
E_{u,y}^- &= (-e_{u,t_y}^{1,t_1} \lor -e_{u,t_y}^{2,t_2} \lor \ldots \lor -e_{u,t_y}^{r,t_r}) \lor e_{u,t_y}^{s,t_z} \\
\end{align*}
\tag{13}
\]

where \( E_{u,x}^+ \) and \( E_{u,y}^- \) are the expressions of the ground-truth interaction and the negative sampled interaction, respectively. Notably, both above logical expressions fuse the temporal information into interactions. Consequently, the truth evaluations of these expressions depend on not only the interacted items but also timestamp, e.g., \( E_{u,x}^+ \) may be false if timestamp is not \( t_x \), while \( E_{u,y}^- \) would be true if timestamp is not \( t_y \).

To evaluate the truth of expressions in Eq.(13), we define \( s_{u,x}^+ \) and \( s_{u,y}^- \) to be the prediction scores for a positive sample of item \( v_x \) at timestamp \( t_x \) and a negative sample of item \( v_y \) at timestamp \( t_y \), respectively. Simulated by LNN layer, the expressions in Eq.(13) is further evaluated in term of similarity via Eq.(12), giving the prediction scores below:

\[
\begin{align*}
s_{u,x}^+ &= \text{Sim}(\text{LNN}(e_{u,t_x}^{1,t_1}, e_{u,t_x}^{2,t_2}, \ldots, e_{u,t_x}^{r,t_r}, e_{u,t_x}^{s,t_z}), T) \\
s_{u,y}^- &= \text{Sim}(\text{LNN}(e_{u,t_y}^{1,t_1}, e_{u,t_y}^{2,t_2}, \ldots, e_{u,t_y}^{r,t_r}, e_{u,t_y}^{s,t_z}), T) \\
\end{align*}
\]

In order to guarantee the TiSANCNR to encourage the prediction score of positive sample and suppress that of negative sample, i.e., to jointly maximize \( s_{u,x}^+ \) and minimize \( s_{u,y}^- \), following NCR [1], we also construct the difference of these two prediction scores as \( s_{v_x,v_y}^+ = \beta \cdot (s_{u,x}^+ - s_{u,y}^-) \), where \( \beta \) is the amplification factor. In this scenario, the objective of model training turns to be maximization of this difference. The difference \( s_{v_x,v_y}^+ \) is further employed to derive the objective function of pair-wise learning, formally,

\[
L_{\text{pair-wise}} = - \sum_{u \in U} \sum_{v_x \in \mathcal{V}_u^+} \sum_{v_y \in \mathcal{V}_u^-} \ln \sigma(s_{v_x,v_y}^+) + \lambda_D ||\Delta||_2^2, \tag{14}
\]

where \( \Delta \) is all parameters of our model, \( \lambda_D \) is coefficient of \( \ell_2 \)-norm regularization which prevents the weights from being too large, and \( \sigma(\cdot) \) is the sigmoid function: \( \sigma(x) = \frac{1}{1+e^{-x}} \).

Note that the LNN layer which simulates the logical expression in the space of event vectors, is a plain neural architecture expected to really realize logical reasoning. To this end, logical regularizer was introduced to constrain the logical neural modules. Integrating logical regularizer into the objective function gives the final loss function:

\[
L = L_{\text{pair-wise}} + \lambda_r L_{\text{logicReg}}, \tag{15}
\]

where \( L_{\text{logicReg}} \) is logic regularizer representing several logical laws, such as negation and double negation in NOT module, identity, annihilator, idempotence and complementation in OR module [1], and \( \lambda_r \) is coefficient for logic regularizer.

It is worth mentioning that in terms of logical regulation, logical regularizer in NCR [1] involves logical expression of the fused embeddings of non-temporal user-item interactions, while our TiSANCNR model deals with that of self-attentive temporal interactions, leading to deeper representation of logical constrains for reasoning-based recommendation.

### III. Experiments

In order to validate the effectiveness of integrating self-attentive temporal information into reasoning-based recommendation, in this section, we conduct experiments on benchmark datasets to demonstrate the advantage of proposed TiSANCNR model over other state-of-the-art recommendation models. In particular, we answer the following research questions from experimental point of view:

- **RQ1:** What’s the performance of our proposed TiSANCNR model compared to other state-of-the-art methods for recommendation systems?
• RQ2: Does the integration of temporal information make a significant impact on reasoning-based recommendation? Furthermore, for the representation of temporal information, which is the better choice, relative time or absolute time?

• RQ3: What is the impact of self-attention on the model performance when considering temporal patterns in reasoning-based recommendation?

• RQ4: What are the effects of hyper-parameters on the performance of TiSANCNR model?

A. Experimental Setup

Datasets: We evaluate our proposed TiSANCNR model and compare it with several baselines by conducting experiments on three public real-world datasets: MovieLens 100K [5], Amazon Movies&TV and Amazon Electronics [13]. The statistics of these three datasets are summarized in Table I.

| Dataset          | #Users | #Items | #Interaction | Density |
|------------------|--------|--------|--------------|---------|
| MovieLens100K    | 943    | 1,682  | 100,000      | 6.3%    |
| Movies&TV        | 123,961| 50,053 | 1,697,533    | 0.027%  |
| Electronics      | 192,404| 63,002 | 1,689,188    | 0.014%  |

We follow the same procedure of data preprocessing from [1]. For implicit feedback, we consider interactions between user and item without ratings to build the logical expression in Eq.(1). While for explicit feedback, we transform the ratings which range from 1 to 5, to 0 and 1. In particular, we treat the ratings of 4 and 5 as positive feedback which are set to 1, and those of 1 to 3 as negative feedback which are set to 0. Given $e_{u,t}$, we select the most recent 5 historical interactions to construct the logical expression, in Eq.(1) with implicit feedback or Eq.(3) with explicit feedback.

For each user, events from earliest 5 interactions and those with less than 5 interactions are directly put into training set. For partitioning of each dataset, we conduct leave-one-out operation on the sequence of historical interactions for each user to create test set and validation set. The most recent event and the second most recent events of interaction are assigned to the test set and validation set respectively, while all remaining events are assigned to training set. Notice that training set and validation set would be leveraged to build logical expression during test phase.

Baselines: We compare our model with the following baseline algorithms, including BPR-MF [14], SVD++ [9], DMF [19], NCF [6], GRU4Rec [7], STAMP [12] and NCR [1]

In addition, we test three versions of our model to prove the effectiveness of integrating temporal information and self-attention in reasoning-based recommendation.

• TiSANCNR-A w/o SA Absolute Time-aware Self-Attention with Neural Collaborative Reasoning without Self-Attention, which use absolute timestamps as temporal information.

• TiSANCNR-R w/o SA Relative Time-aware Self-Attention with Neural Collaborative Reasoning without Self-Attention, which use relative timestamps as temporal information.

Metrics: We employ two widely used ranking based metrics, namely Normalized Discounted Cumulative Gain at rank $k$ (NDCG@$k$) and Hit Ratio at rank $k$ (HR@$k$).

In experiments, we employ the popular setting of $k = 5, 10, 20$ for validation, and the result of both metrics are averaged over all users to reduce the effect of random oscillation. To avoid heavy computational burden, for each user, 100 negative samples are randomly drawn and mixed with the ground-truth (i.e., positive sample) in the ranking process.

Implementation details: The parameters for all baselines were initialized as in the corresponding papers, and carefully tuned to obtain the optimal performance. The implementation details of TiSANCNR are provided as follows. The dimensions of users, item, and event embedding latent vectors are determined by grid search in the range of $\{20, 40, 60, 80, 100\}$. The weight of logical regularizer $\lambda$ in Eq.(15) is set to 0.01. For model training, the batch size of training examples is set to 128, and the maximum of training epoch is determined by grid search in the range of $\{50, 60, 80, 100, 200, 500, 1000\}$. Both $\ell_2$ regularization and dropout are employed to prevent model fitting. The weight of $\ell_2$ regularization and dropout ratio are set to $10^{-3}$ and 0.2, respectively. All of the parameters are initialized according to Gaussian distribution $\mathcal{N}(0, 0.01)$, and updated with Adam optimizer. The learning rate of Adam is set to $10^{-3}$. All experiments on benchmark datasets are implemented with PyTorch, on a 64 core Intel Xeon Gold 6226R CPU @ 2.90GHz, 256 GB memory and a Nvidia Quadro RTX 8000 GPU.

B. Comparison of Overall Performance

Table II reports the comparisons w.r.t. metrics NDCG(N) and HR at rank 5 and 10 on all of the datasets.

We first analyze the recommendation results of the baselines. As sequence/session recommendation models, GRU4Rec and STAMP mostly achieve best performance than the first four models upon all three datasets in terms of all metrics. We can see that GRU4Rec and STAMP are the best among matching-based recommendation models. On the other hand, NCR obtains the best performance among the baselines in terms of all metrics, which implies that the integration of perception learning and cognitive reasoning into recommendation task provides significant improvement on the ranking performance. Hence, we refer NCR to the best result of baselines for comparisons in further experiments.

Next we evaluate our proposed TiSANCNR model against the baselines in terms of the ranking performance. As shown in Table II, the numbers with bold fonts indicate that our proposed model TiSANCNR consistently and noticeably outperforms NCR, i.e., the best result among baselines on all of the datasets. This shows that the incorporation of temporal information and self-attention mechanism into reasoning-
basing recommendation helps to improve the recommendation performance. In particular, significant improvements against NCR are observed in Table II. The significant improvement may be due to the incorporation of temporal information and self-attention mechanism, which motivates us to further investigate the influences of temporal information and self-attention mechanism on reasoning-based recommendation model to answer RQ2 and RQ3.

C. Influence of Absolute and Relative Time Pattern

There are two typical temporal patterns in recommendation system, including absolute and relative time pattern. In order to investigate the pros and cons of these two temporal patterns in reasoning-based recommendation, we conduct ablation experiments on three datasets mentioned above.

When considering absolute time pattern to be temporal context, we extract absolute timestamps in each user-item interaction and fed them into LNN layer without self-attention layer in network architecture, leading to the TiSANCR-A w/o SA model. In Table II, the eighth row shows the ranking performance on three datasets, and Improvement\(^2\) row reports the improvement of TiSANCR-A w/o SA over NCR. Compared with NCR, the TiSANCR-A w/o SA model obtains little even no improvement on the ranking performance. Hence, leveraging absolute time pattern can not guarantee improvement to reasoning-based recommendation, sometimes would ruin the improvement of model performance.

On the other hand, we turn to integrate relative time pattern to reasoning-based recommendation. In network architecture, we extract relative times, and also fed them into LNN layer without self-attention layer, resulting in the TiSANCR-R w/o SA model. In Table II, the ninth row shows the ranking performance of the TiSANCR-R w/o SA model on three datasets, and Improvement\(^3\) row reports the improvement of TiSANCR-R w/o SA over NCR. We can observe that the ranking performance of the TiSANCR-R w/o SA model is superior to those of baselines and the TiSANCR-A w/o SA model. It is not surprising that relative time pattern probably provides more highlighted relevance between historical interactions and candidate item at current timestamp. Improvement\(^4\) row manifests the improvement of TiSANCR-R w/o SA model over of TiSANCR-A w/o SA model, which also confirms that relative time patterns are extremely important for reasoning-based recommendation. Compared with the best baseline NCR, the integration of relative time patterns in reasoning-based recommendation shows significant superiority to model the temporal dynamics of user’s preference, hence improving the effectiveness of logical reasoning in recommendation.

D. Influence of Self-Attention

As mentioned above, integration of relative time patterns in reasoning-based recommendation provides grate improvement of ranking performance, however the model of RS would tend to be self-attentive on temporal information. Here we investigate the impact of self-attention mechanism on the ranking performance of reasoning-based recommendation incorporated with relative time patterns. In order to answer RQ3, we mainly focus on discussing two variations of recommendation models, namely TiSANCR-R w/o SA and TiSANCR, where self-attention mechanism is (not) leveraged to distill informative temporal patterns and suppress irrelevance in TiSANCR (TiSANCR-R w/o SA) model, respectively.

In Table II, Improvement\(^5\) row indicates the improvement of proposed TiSANCR over TiSANCR-R w/o SA. It can be observed that employing self-attention mechanism to highlight the importance of temporal patterns helps further improve the
ranking performance. Such observation demonstrates the effectiveness of self-attention to extract the importance of relative time patterns. It is interesting to see that the improvements of leveraging self-attention are slight compared with that of integrating relative time patterns. The reason may be that during training, the weights of model with temporal information gradually concentrates on most relevance of temporal patterns, and the self-attention is further employed to encourage the model to distill informative patterns and suppress irrelevance.

E. Ablation Study on Hyper-parameters

In order to answer RQ4, in this section, we investigate the impacts of hyper-parameters via an ablation study. The hyper-parameters include the number of dimensions of users, item, and event embedding latent vectors, and the number of training epoch. We also investigate the ranking performance of proposed model TiSANC R over the metrics HR and NDCG at different rank, i.e., HR@k and NDCG@k.

**Embedding dimension:** To determine the effect of dimension of embedding latent vectors, we conduct a comparative experiment on the proposed TiSANC R. We focus on discussing the number of embedding vectors of users, items and the fusion of events. We report the effect of latent dimensionality on the ranking performance over the three datasets. Fig. 3 demonstrates the validation performance under different metrics including HR@5, HR@10, NDCG@5 and NDCG@10, w.r.t. different number of embedding dimension.

Comparing the results under different metrics, we find that our model typically benefits from larger number of dimensionality in most cases, and achieve satisfactory performance with $d \geq 60$. Furthermore, Fig. 3 shows that TiSANC R mostly achieves the best performance on MovieLens100K dataset, and better performance on Movie&TV dataset, compared with the performance on Electronics dataset. This finding reports that TiSANC R typically obtains better on dense dataset with moderate size. Comparing the upper and lower sides in Fig. 3, we can also observe that the performance of TiSANC R in terms of HR are more superior than those of NDCG.

**Number of epoch:** In order to investigate the effect of training epoch on the ranking performance of TiSANC R, we conduct experiments on the training of TiSANC R with different numbers of training epochs, which is selected in range of 5 to 1,000. Fig. 4 shows the ranking performance over the three datasets across different numbers of training epochs. In Fig. 4, we can see that as the number of training epoch increases, the recommendation performance of TiSANC R over all datasets is gradually improved and then achieves convergence. The oscillations of performance results is due to the randomness of sampling in model training.

For MovieLens100K dataset, the ranking performance converges to 0.6 over HR@5, 0.73 over HR@10, 0.45 over NDCG@5 and 0.5 over NDCG@10. For Electronics dataset, the ranking performance converges to 0.52 over HR@5, 0.65 over HR@10, 0.4 over NDCG@5 and 0.43 over NDCG@10. For Movie&TV dataset, the ranking performance achieves best result of 0.55, 0.73, 0.4 and 0.45 over HR@5, HR@10, NDCG@5 and NDCG@10 respectively, and then degrades after reaching the peak. For all datasets, TiSANC R obtains satisfactory performance with training epoch larger than 100.

**Metrics at different rank:** We consider to investigate the recommendation performance of TiSANC R under HR and NDCG at different rank $k$, where $k$ ranges from 5 to 20. Fig. 5 manifests the evaluation performance of proposed TiSANC R under metrics HR and NDCG. It is obvious that, with the size of rank $k$ list ranging from 5 to 20, the overall
The performance of recommendation increases, which is consistent with the conclusion in the literature of RS. Comparing the results in Fig. 5(a) and (b), the performance of TiSANCR in terms of HR are more superior than those in terms of NDCG. In Fig. 5(b), under NDCG metric with different rank, the recommendation performance over MovieLens100K dataset is better than those over other two datasets, which also indicates that TiSANCR typically obtains better on dense dataset with moderate size. In Fig. 5(a), we can observe that under HR metric with different rank, the performance of TiSANCR over Electronics dataset is worse than those of other two datasets.

IV. CONCLUSION

In this paper, we aim to leverage the temporal information and self-attention mechanism to capture deeper representation of user’s preference for recommender system. A novel recommendation model named TiSANCR is proposed, where relative time patterns is employed to provide more context information, while self-attention is adopted to capture informative temporal context and suppress irrelevant patterns in historical interactions, leading to better representation of the temporal information for candidate item recommendation. We conduct extensive experiments on three benchmark datasets to evaluate the effectiveness of TiSANCR on modeling temporal patterns. The experimental results demonstrate that our TiSANCR outperforms the state-of-the-art recommendation approaches.

ACKNOWLEDGMENT

This work was supported by the Teaching Quality Engineering Construction Project of Jiaying University in 2022 under Grant YYG2022239, the Research Project of Jiaying University under Grant 2023RC60, the 2023 Teaching Quality Project for Undergraduate Institutions of Guangdong Province, and Special Funds for the Cultivation of Guangdong College Students’ Scientific and Technological Innovation (“Climbing Program” Special Funds) under Grant pdjh2023b0486.

REFERENCES

[1] Hanxiong Chen, Shaoyun Shi, Yunqi Li, and Yongfeng Zhang. Neural collaborative reasoning. In Proceedings of the Web Conference 2021, pages 1516–1527, 2021.

[2] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Janneck. A troubling analysis of reproducibility and progress in recommender systems research. ACM Trans. Inf. Syst., 39(2):1–49, 2021.

[3] Maurizio Ferrari Dacrema, Federico Parroni, Paolo Cremonesi, and Dietmar Janneck. Critically examining the claimed value of convolutions over user-item embedding maps for recommender systems. In Proc. ACM Int. Conf. Inf. Knowl. Manag., pages 355–363, 2020.

[4] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. Proc. IDC iView: IDC Analyze the future, 2007(2012):1–16, 2012.

[5] F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst., 5(4):1–19, 2015.

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In Proc. Int. Conf. World Wide Web, pages 173–182, 2017.

[7] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939, 2015.

[8] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In Proc. IEEE Int. Conf. Data Mining, pages 197–206, 2018.

[9] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In Proc. Int. Conf. Knowl. Discov. Data Mining, pages 426–434, 2008.

[10] Yehuda Koren. Collaborative filtering with temporal dynamics. In Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pages 447–456, 2009.

[11] Jiacheng Li, Yujie Wang, and Julian McAuley. Time interval aware self-attention for sequential recommendation. In Proc. Int. Conf. Web Search Data Mining, pages 322–330, 2020.

[12] Qiao Liu, Yifu Zeng, Refuoe Mokhosi, and Haibin Zhang. Stamp: short-term attention/memory priority model for session-based recommendation. In Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, pages 1831–1839, 2018.

[13] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In Proc. Int. ACM SIGIR Conf. Res. Dev. Info. Retr., pages 43–52, 2015.

[14] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618, 2012.

[15] Suvalash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. AutoRec: Autoencoders meet collaborative filtering. In Proc. ACM Int. Conf. World Wide Web, pages 111–112, 2015.

[16] Shaoyun Shi, Hanxiong Chen, Weizhi Ma, Jiuxin Mao, Min Zhang, and Yongfeng Zhang. Neural logic reasoning. In Proc. ACM Int. Conf. Inf. Knowl. Manag., pages 1365–1374, 2020.

[17] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In Proc. Int. Conf. Knowl. Discov. Data Mining, pages 197–206, 2015.

[18] Yao Wu, Christopher DuBois, Alice X Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In Proc. ACM Int. Conf. Knowl. Discov. Data Mining, pages 153–162, 2016.

[19] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. In Int. Joint Conf Artif. Intell., volume 17, pages 3203–3209, Melbourne, Australia, 2017.

[20] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. Time matters: Sequential recommendation with complex temporal information. In Proc. Int. ACM SIGIR Conf. Res. Dev. Info. Retr., pages 1459–1468, 2020.

[21] Lei Zheng, Vahid Noroozi, and Philip S Yu. Joint deep modeling of users and items using reviews for recommendation. In Proc. Int. Conf. Knowl. Discov. Data Mining, pages 425–434, 2017.