

# Extracting Attentive Social Temporal Excitation for Sequential Recommendation

Yunzhe Li<sup>1</sup>, Yue Ding<sup>1</sup>, Bo Chen<sup>2</sup>, Xin Xin<sup>3</sup>  
Yule Wang<sup>1</sup>, Yuxiang Shi<sup>1</sup>, Ruiming Tang<sup>2</sup>, Dong Wang<sup>1†</sup>

<sup>1</sup>Shanghai Jiao Tong University, <sup>2</sup>Huawei Noah's Ark Lab, <sup>3</sup>University of Glasgow, <sup>†</sup>Corresponding authors  
{liyhz28,dingyue,wyl666,shiyuxiang,wangdong}@sjtu.edu.cn  
{chenbo116,tangruiming}@huawei.com,x.xin.1@research.gla.ac.uk

## ABSTRACT

In collaborative filtering, it is an important way to make full use of social information to improve the recommendation quality, which has been proved to be effective because user behavior will be affected by her friends. However, existing works leverage the social relationship to aggregate user features from friends' historical behavior sequences in a user-level *indirect paradigm*. A significant defect of the indirect paradigm is that it ignores the temporal relationships between behavior events across users. In this paper, we propose a novel time-aware sequential recommendation framework called Social Temporal Excitation Networks (STEN), which introduces temporal point processes to model the fine-grained impact of friends' behaviors on the user's dynamic interests in an event-level *direct paradigm*. Moreover, we propose to decompose the temporal effect in sequential recommendation into social mutual temporal effect and ego temporal effect. Specifically, we employ a social heterogeneous graph embedding layer to refine user representation via structural information. To enhance temporal information propagation, STEN directly extracts the fine-grained temporal mutual influence of friends' behaviors through the *mutually exciting temporal network*. Besides, user's dynamic interests are captured through the *self-exciting temporal network*. Extensive experiments on three real-world datasets show that STEN outperforms state-of-the-art baseline methods. Moreover, STEN provides event-level recommendation explainability, which is also illustrated experimentally.

## CCS CONCEPTS

• Information systems → Recommender systems; • Computing methodologies → Neural networks.

## KEYWORDS

Temporal Point Process; Sequential Recommendation; Attention Mechanism; Social Recommendation

## ACM Reference Format:

Yunzhe Li<sup>1</sup>, Yue Ding<sup>1</sup>, Bo Chen<sup>2</sup>, Xin Xin<sup>3</sup> and Yule Wang<sup>1</sup>, Yuxiang Shi<sup>1</sup>, Ruiming Tang<sup>2</sup>, Dong Wang<sup>1†</sup>. 2021. Extracting Attentive Social Temporal Excitation for Sequential Recommendation. In *Proceedings of the 30th*

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 978-1-4503-8446-9/21/11...\$15.00  
<https://doi.org/10.1145/3459637.3482257>

ACM International Conference on Information and Knowledge Management (CIKM'21), November 1–5, 2021, Virtual Event, QLD, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3459637.3482257>

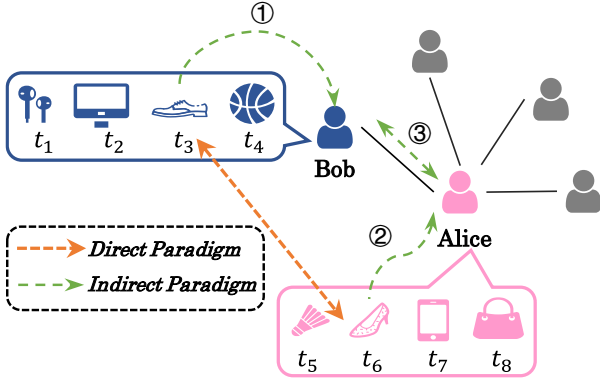
## 1 INTRODUCTION

In recent years, sequential recommendation, as an emerging topic for capturing user dynamic interests from user behaviors, has received extensive attention from the industry and academia. Various methods emerge to utilize deep sequential models, such as Recurrent Neural Networks (RNNs) [9], Convolution Neural Networks (CNNs) [32], Attention mechanisms [11, 14] to extract features from the successive behavior event sequence and enhance recommendation performance. Moreover, some works [1, 15] model the time interval to capture users' demands over time. **Nevertheless, for an enormous set of items, only a tiny fraction is involved in a single user behavior sequence. This phenomenon limits the performance of sequential recommendation.**

Furthermore, **based on the social influence and social homophily [19, 20, 27], social relationship plays an important role in recommending suitable items for users.** Some works leverage the social relations to aggregate similar users' preferences to enhance collaborative filtering and matrix factorization or integrate deep neural networks into probabilistic matrix factorization for rating prediction [5, 18, 31, 34, 43]. Moreover, Graph Neural Networks (GNNs) are also utilized to enhance the performance of recommendation for better capturing the structural information of social relations [6, 22, 38, 39]. **However, these approaches are coarse-grained that only model user-level relationships and ignore the influence of the user behavior sequence.**

To leverage social influence to enhance sequential recommendation performance, existing methods [28, 30] usually model both social relations and user's behavior sequences in an indirect manner. As Figure 1 shows, the *indirect paradigm* of the methods (the green path) usually can be summarized in three steps: (1) learning friends' representation from their behavior sequences; (2) learning the target user's representation from her own behavior sequence; (3) aggregating the influence of friends on the target user through their representations. However, the indirect paradigm usually suffers from information loss due to the ignorance of the relationships between behavior events across users. Compared with the indirect paradigm, modeling the influence of different user behavior sequences in the *direct paradigm* (the orange path) is more granular to avoid the problem mentioned above and better model the interaction between cross-user events.

Nonetheless, to the best of our knowledge, existing works do not contemplate such a direct modeling paradigm, leaving vast scope



**Figure 1: An illustration of two paradigms to deal with asynchronous behavior event sequences and social relations in online communities.**

for exploration. Besides, unlike the synchronous sequence taken at consecutive equal intervals, the users' behavior sequences tend to be asynchronous when considering interactions across user events. Different from only taking the order of events into consideration, how to leverage the temporal information is another challenge for properly modeling the asynchronous sequences. We use a simple example shown in Figure 1 to illustrate this issue. If *Alice* buys heels within a short time after her friend *Bob* buys dress shoes, i.e.,  $t_3 \leq t_6$ , then it indicates that *Bob*'s behavior is more likely to affect her (e.g., attending certain common activities). But if  $t_3 \ll t_6$  or  $t_3 > t_6$ , *Bob*'s behavior will have less or no effect on *Alice*. If the temporal information is lost, the model will not be able to determine the specific relationship between the two events. Hence, the time interval should be treated as important as the order, especially when we note the interaction of cross-user events.

To address the problems mentioned above, we propose a novel approach called **Social Temporal Excitation Networks (STEN)**, which models individual user's sequence and the interaction between users together. STEN is designed to learn the influence of temporal information by modeling the temporal effect, which is decomposed into social mutual temporal effect and ego temporal effect. Specifically, the former aggregates the impact of friends' behavior sequences over time, and the latter describes the changes in interests reflected by the user's own behavior sequence. Our model introduces the temporal point process, which is a powerful tool for asynchronous sequence modeling, to model the effect of related cross-user events with temporal information in a fine-grained manner. Specifically, node representation of both users and items are extracted from the social relationship structure and interactions by a *social heterogeneous graph embedding layer*. To explore the impact of cross-user events in a direct paradigm, we model the interaction of friends' behavior event sequences as mutually exciting processes and enhance it by attention mechanism to better describe the social mutual temporal effect in the *mutually exciting temporal network*. Finally, a *self-exciting temporal network* is employed to model the target user's ego temporal effect and generate the user's temporal representation. To summarize, we make the following contributions in this paper:

- We propose to take temporal information and social relationships into account in the sequential recommendation, i.e., model the impact of friends' behavior event sequences at the event level. To the best of our knowledge, it is the first attempt to enhance sequential recommendation performance with event-level social information in a direct paradigm.
- We propose a novel time-aware sequential recommendation model STEN. Two temporal excitation networks are designed for modeling both the event-level dynamic influence of social relationship and user's dynamic interests. In particular, STEN can model the fine-grained associations of cross-user events.
- We conduct extensive experiments on three real-world datasets and the results demonstrate that STEN outperforms state-of-the-art recommendation models and the effectiveness of the modules in STEN. Besides, we illustrate the advantage of the event-level explainability of our model.

## 2 PRELIMINARY

In this section, we first introduce the temporal point process as background and then give the problem formulation.

### 2.1 Temporal Point Process

A temporal point process is a stochastic process consisting of a series of events in the continuous-time domain. Let  $H_t = \{t_i | t_i < t\}$  be the set of historical events that happened before time  $t$ , where  $\{t_1, t_2, \dots, t_n\}$  is the time sequence of past events. To characterize the Point Process  $\mathcal{P}(\lambda(t|H_t))$  for arbitrary time  $t$  in a certain time interval  $[0, T]$ , a conditional intensity function  $\lambda(t|H_t)$  is imported to represent the intensity function of the event occurring at time  $t$  under the condition of the subset of history events  $H_t$  up to time  $t$ . Based on the conditional intensity function and the past events in  $\{t_1, t_2, \dots, t_n\}$ , the probability density function of the next event at time  $t_{n+1}$  is obtained as follows.

$$p(t_{n+1}|t_1, t_2, \dots, t_n) = \lambda(t_{n+1}|H_{t_{n+1}}) \exp\left\{-\int_{t_n}^{t_{n+1}} \lambda(t|H_t) dt\right\}. \quad (1)$$

The probability density function of a whole observed event sequence occurred at  $\{t_1, t_2, \dots, t_n\}$  in interval  $[0, T]$  could be represented by:

$$p(\{t_1, t_2, \dots, t_n\}) = \prod_{i=1}^n \lambda(t_i|H_{t_i}) \exp\left\{-\int_0^T \lambda(t|H_t) dt\right\}. \quad (2)$$

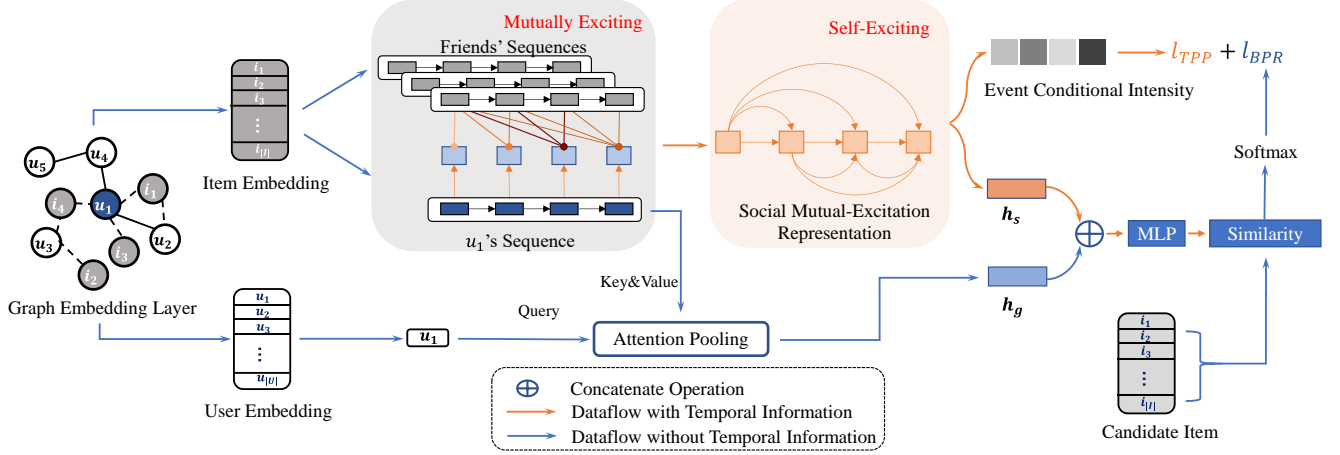
The log-likelihood function  $\ell$  of the observing event sequence is:

$$\ell(p(\{t_1, t_2, \dots, t_n\})) = \sum_{i=1}^n \log(\lambda(t_i|H_{t_i})) - \int_0^T \lambda(t|H_t) dt. \quad (3)$$

Denote as Eq. (3), the conditional intensity function plays an essential role in mapping the history event sequence to the probability density function. Take the Hawkes Process [8] as an example, the conditional intensity function of this popular point process is denoted as Eq. (4):

$$\lambda(t|H_t) = \sum_{t_i > t_i} \varphi(t - t_i), \quad (4)$$

where  $\varphi(\cdot)$  is a non-negative function. Eq. (4) shows that the historical events have an incentive effect on the occurrence of future events and the influence of historical events is superimposed in



**Figure 2: The architecture of STEN.** Firstly, through the social graph embedding layer, the node representation of users and items is generated by leveraging the sequence of events and social relations. Then, the embeddings of events are sent to the mutually exciting attention networks to obtain the social mutual temporal representations. Next, mutual temporal representations are fused in self-exciting temporal networks to capture the temporal dynamic representations  $h_s$ . Besides, vanilla attention is adopted to get the general interests  $h_g$ . After that,  $h_s, h_g$  are fused to make recommendation. In addition, here adopts joint learning for both fitting the temporal point process and recommendation.

the form of accumulation. Based on whether there are interactions between multiple processes, the Hawkes Process could be divided into mutually exciting and self-exciting processes.

## 2.2 Problem Formulation

The time-aware sequential recommendation with social information takes both user behavior sequences and their social relations into consideration. Let  $U, I$  denote the set of users and items respectively and  $|U|, |I|$  represent the total number of users and items. For each user  $u \in U$ , her behavior sequence is described as  $\mathbf{s}_u = \{i_{t_1}^u, i_{t_2}^u, \dots, i_{t_l}^u\}$ , where  $l$  is the length of sequence and  $i_{t_j}^u \in I$  denotes the item that user  $u$  has interacted at time  $t_j$ . The set of all users' behavior sequences is defined as  $S = \{s_1, s_2, \dots, s_{|U|}\}$ . For a subset of users  $F \subset U$ , the set of their behavior sequences is denoted as  $S_F$ . In addition, user's social relationship could be defined as a social network  $G = (V, E)$ , where  $V$  denotes the node set of users and  $E$  is the edge set that represents social links between users. Formally, given users' social relations network  $G$  and their behavior sequences  $S$ , our task is to make a prediction and recommend the next item that the user may be interested in.

## 3 METHODOLOGIES

As mentioned above, to take the influence of both behavior sequences and social relations into consideration, we design a novel time-aware sequential recommendation model STEN, which introduces temporal point processes to directly model the fine-grained impact of friends' behaviors at the event level.

### 3.1 Model Architecture

The architecture of our model STEN is shown in Figure 2. Specifically, we leverage social relations and user behavior sequences to generate the social node embedding for both users and items through the social graph embedding layer. To capture the ordinary

social demands, vanilla attention is adopted to get the general interests of the target user. To directly model the impact of friends' behavior sequences, the embeddings of their interacted sequences are sent to the mutually exciting temporal network as mutually exciting processes to generate the social mutual temporal representations. To establish the connection within the sequences, these social mutual temporal representations are fused in the self-exciting temporal network to capture the target user's dynamic interests. After that, we fuse the general interests and the dynamic interests to make recommendations. To guarantee effective learning from the temporal point process, we adopt joint learning in the training stage for both fitting the temporal point process and recommendation.

**3.1.1 Social Heterogeneous Graph Embedding.** The social graph embedding aims to extract the general information from the social relations and interactions. In our model, we leverage a heterogeneous graph to represent the relationships for better scalability. For a social heterogeneous network  $G_H = (V_H, E_H)$ ,  $V_H$  denotes the nodes of both items and users,  $E_H$  denotes the edge set which involves two types of edges: (1) user-user social links; (2) user-item interaction connection. We first generate the embedding matrices  $U \in \mathbb{R}^{|U| \times d}$ ,  $I \in \mathbb{R}^{|I| \times d}$  via embedding lookup, where  $d$  denotes the dimensionality of embedding. To tackle the two types of edge, we give two kinds of aggregation to model both users and items. For users, we could directly establish the social relation network through user-user edges and then apply attention aggregation on it, shown as Eq.(5):

$$h_u = \sigma\left(\sum_{j \in N_u} a_{uj} \mathbf{W}_U h_j + b_U\right), \quad (5)$$

where  $h_u \in \mathbb{R}^d$  is latent feature of user  $u$ ,  $\sigma(\cdot)$  is the non-linear activation function,  $N_u$  denotes the neighbor set of user  $u$ ,  $\mathbf{W}_U \in \mathbb{R}^{d \times d}$  and  $b_U \in \mathbb{R}^d$  are weights and bias of the graph network,  $a_{uj}$  is the attention coefficient of user  $u$  to user  $j$ . The attention

coefficient is obtained by applying a similarity calculation:

$$a_{uj} = \frac{\exp(\sigma(v_u[\mathbf{W}_U h_u || \mathbf{W}_U h_j]))}{\sum_{k \in N_u} \exp(\sigma(v_u[\mathbf{W}_U h_u || \mathbf{W}_U h_k]))}, \quad (6)$$

where  $||$  denotes the concatenation operation between two vectors and  $v_u \in \mathbb{R}^{2d}$  is the weight vector for users.

Different from user links, there is no direct relationship between items. To establish the relation, we leverage meta-path to find high-order neighbors around the heterogeneous graph[35]. If two items are interacted with the same user, they are more likely to have something in common that satisfies user's interests. Moreover, the items interacted by the same social group are more likely to have relationship. Based on the assumption, we define the  $k$ -order social meta-path  $\phi_k$  as  $\{i_m, u_1, u_2, \dots, u_k, i_n\}$ , which denotes a combination relation between item  $i_m$  and  $i_n$ ,  $k$  is the number of users between two items.  $N_i^{\phi_k}$  denotes the  $k$ -order meta-path neighbors of item  $i$ . In the next, the attention aggregation is applied to the item vertices, shown as follows:

$$h_i^{\phi_k} = \sigma(\sum_{j \in N_i^{\phi_k}} a_{ij} \mathbf{W}_I h_j + b_I), \quad (7)$$

where  $h_i^{\phi_k}$  is the  $k$ -order representation of item  $i$ ,  $\mathbf{W}_I \in \mathbb{R}^{d \times d}$  and  $b_I \in \mathbb{R}^d$  are weight and bias parameters of the graph network. Similar to Eq. (6), the calculation of  $a_{ij}$  is shown as Eq. (8), where  $v_i \in \mathbb{R}^{2d}$  is the weight vector for items.

$$a_{ij} = \frac{\exp(\sigma(v_i[\mathbf{W}_I h_i || \mathbf{W}_I h_j]))}{\sum_{p \in N_i^{\phi_k}} \exp(\sigma(v_i[\mathbf{W}_I h_i || \mathbf{W}_I h_p]))}. \quad (8)$$

In addition, an order-level normalization is applied to items to obtain the graph-based representation:

$$h_i^\phi = \frac{1}{l} \sum_{k \leq l} c_k h_i^{\phi_k}, \quad (9)$$

where  $h_i^\phi$  denotes the meta-path based representation of item  $i$ ,  $l$  is the number of meta-paths,  $c = \{c_1, \dots, c_k\}$  are weight parameters. We replace  $h_i^{\phi_k}$  by  $h_i$  in the rest of the paper for convenience.

**3.1.2 Mutually Exciting Temporal Network.** To capture temporal features and the fine-grained influence of social relations, STEN imports two neural temporal point processes: a mutually exciting process to model the influence of friends' behaviors from each event and a self-exciting process to model user's dynamic interests. Different from the RNN-based neural point process model, STEN utilizes the attention mechanism, which could avoid the disappearance of long-distance dependence.

Specifically, STEN divides the mission into two steps: 1) Model the mutual influence of other social-connected users with the target user through mutually exciting temporal network; 2) Extract the user temporal feature from her own sequence through self-exciting temporal network.

Firstly, we model the target user  $n$  and friends' event sequences as temporal point processes, denote as  $\mathcal{P}_n(\lambda(t|H_t))$ ,  $\{\mathcal{P}_j(\lambda(t|H_t))\}_{j=1}^M$ , where  $M$  is the maximum number of friends.  $|\mathcal{P}_j|$  denotes the length of event sequence for user  $j$ . In each temporal point process, an event  $e_t^j$  means that user  $j$  has an interactive behavior with item at

time  $t$ . To measure friends' influence for each event timestamp in target behavior sequence, STEN models a mutually exciting process for the target user, defined as:

$$\lambda_n^m(t_i|H_{t_i}^n) = \alpha \sum_{j=1}^M \int_0^{t_i} \lambda_{n,j}(t_i, t|H_{t_i}^n, H_t^j) dt, \quad (10)$$

where  $H_t^j$  is the condition of user  $j$ 's behavior history before time  $t$ ,  $\lambda_{n,j}(t_i, t)$  represents the mutual influence intensity of friend  $j$  at time  $t$  on the target user at time  $t_i$ .  $\alpha$  is a normalized parameter. For users whose friends exceed  $M$ , we adopt a simple sampling strategy to randomly select  $M$  friends. Because Eq. (10) explores the impact on the target user, we assume that the friends' events are independent of each other. Based on that assumption, the mutually exciting temporal network could treat each event separately. The mutual influence intensity of each friend can be regarded as a sum of the influence of different events and the mutual influence conditional intensity with two events  $e_{t_i}^n, e_{t_w}^j$  follows the form of:

$$\lambda_n^m(t_i|H_{t_i}^n) = \alpha \sum_{j=1}^M \sum_{w=1}^{|\mathcal{P}_j|} \Phi(e_{t_i}^n, e_{t_w}^j), \quad (11)$$

where  $\Phi(e_{t_i}^n, e_{t_w}^j)$  is the cumulative mutual influence intensity function for events  $e_{t_i}^n, e_{t_w}^j$  and defined as follows:

$$\Phi(e_{t_i}^n, e_{t_w}^j) = \int_{t_w}^{t_i} \lambda_{n,j,w}(t_i, t|H_{t_i}^n, H_t^j) dt. \quad (12)$$

Inspired by[23], we propose to model the cumulative mutual influence intensity function  $\Phi(e_{t_i}^n, e_{t_w}^j)$ , rather than the mutual influence intensity. Compared with the latter, our approach has less limitation in flexible modeling by avoiding costly approximate calculation of integral of mutual influence condition intensity shown in Eq. (10). To take both the content of events and time interval into consideration, the cumulative mutual influence intensity function contains three components: the correlation between two events, the temporal relationship and base intensity, which is defined as follows:

$$\Phi(e_{t_i}^n, e_{t_w}^j) = \underbrace{\mathcal{F}(\text{corr}(e_{t_i}^n, e_{t_w}^j))}_{\text{correlation}} + \underbrace{\beta(t_i - t_w)}_{\text{temporal}} + \underbrace{\mu}_{\text{base}}, \quad (13)$$

where  $\mathcal{F}(\cdot)$  is a mapping function:  $\mathbb{R} \rightarrow \mathbb{R}^+$  to guarantee to obtain the positive cumulative mutual influence intensity, and we use soft-plus here. To consider the influence of friends' events, we concatenate friends' event embedding together as  $\mathbf{S}_F \in \mathbb{R}^{d_F \times d}$ ,  $d_F = M \times l_m$  and  $l_m$  denotes the maximum length of the event sequence. Based on Eq. (11) to (13), we have:

$$A^m = \text{Softmax}(\text{mask}(\frac{\mathbf{Q}_m \mathbf{K}_m^T}{\sqrt{d}} + \beta_m \Delta t + \mu_m))), \quad (14)$$

$$\lambda_n^m(t_i|H_{t_i}^n) = \sum_{k=1}^{d_F} \alpha_k^m \mathbf{W}_m^V h_k^F, \quad (15)$$

where  $\mathbf{Q}_m = h_i^n \mathbf{W}_m^Q$ ,  $\mathbf{K}_m = \mathbf{S}_F \mathbf{W}_m^K$  are query and key matrices transformed by  $h_i^n$  and  $\mathbf{S}_F$ .  $\mathbf{W}_m^Q, \mathbf{W}_m^K, \mathbf{W}_m^V \in \mathbb{R}^{d \times d}$ ,  $\beta_m, \mu_m \in \mathbb{R}$  are

trainable parameters of networks.  $\alpha_k^m \in \mathbf{A}^m$  is the mutual attention score for friends' events.  $h_i^n \in \mathbf{s}_n$ ,  $h_k^F \in \mathbf{S}_F$  are the representations of interactive item of target user and her friends learned from social heterogeneous graph, respectively.  $\Delta t$  denotes the time interval between the friends' events and the target event  $e_{t_i}^n$ . Besides, we apply a mask operation on the temporal attention network to avoid leveraging the "future information". In details, for target event  $e_{t_i}^n$ , we mask all friends' events that happened after time  $t_i$ .

After the process of mutually exciting temporal network, STEN has extracted the influence of friends on the target user's point process, denoted as  $\mathcal{P}_n(\lambda_n^m(t|H_t)) = T_n^m = \{\lambda_{n,i}^m\}_{i=1}^{|\mathcal{P}_n|}$ ,  $\lambda_{n,i}^m$  is the social mutual temporal representation of the  $i$ -th event for user  $n$ . However, there still leaves efforts on modeling the target user's event sequence to obtain her temporal representation.

**3.1.3 Self-Exciting Temporal Network.** The goal of a self-exciting temporal network is to fuse the sequence of temporal representations for interactive items to generate the dynamic representation of users. The main idea of this structure follows the self-exciting process, which is combined with the attention mechanism. For the target process  $\mathcal{P}_n(\lambda_n(t_i|H_{t_i}^n))$ , we apply the attention mechanism to capture the dependency from the sequence without the limitation of length:

$$T_n^S = \text{Softmax}(\text{mask}(\frac{\mathbf{Q}_S \mathbf{K}_S^T}{\sqrt{d}} + \beta_S \Delta t + \mu_S)) \mathbf{V}_S, \quad (16)$$

where  $\mathbf{Q}_S = T_n^m \mathbf{W}_S^Q$ ,  $\mathbf{K}_S = T_n^m \mathbf{W}_S^K$ ,  $\mathbf{V}_S = T_n^m \mathbf{W}_S^V$  are query, key and value matrices obtained by linear transformations of  $T_n^m$ .  $\mathbf{W}_S^Q$ ,  $\mathbf{W}_S^K$ ,  $\mathbf{W}_S^V \in \mathbb{R}^{d \times d}$ ,  $\beta_S, \mu_S \in \mathbb{R}$  are trainable parameters. Similar to the mutually exciting temporal network, we also apply temporal mask operation on the sequence to avoid leveraging "future information". From Eq. (16), we obtain the discrete value of condition intensity function over events  $T_n^S = \{\lambda_{n,i}^S\}_{i=1}^{|\mathcal{P}_n|}$ . We use the output of the last term as the temporal representation of user  $h_s$ .

**3.1.4 Model Prediction.** In addition, to model general social demands, we utilize the target user embedding  $h_n$  learned from social heterogeneous graph as the query and apply a vanilla attention mechanism and average pooling to the target user interacted item sequence  $I^n = \{h_1^n, h_2^n, \dots, h_l^n\}$ , which is also generated by social heterogeneous graph, shown as:

$$h_g = \sum \text{Softmax}(\frac{\mathbf{Q}_g \mathbf{K}_g^T}{\sqrt{d}}) \mathbf{V}_g, \quad (17)$$

where  $\mathbf{Q}_g = h_n \mathbf{W}_g^Q$ ,  $\mathbf{K}_g = I^n \mathbf{W}_g^K$ ,  $\mathbf{V}_g = I^n \mathbf{W}_g^V$  are query, key and value matrices transformed by  $h_n, I^n$ .  $\mathbf{W}_g^Q, \mathbf{W}_g^K, \mathbf{W}_g^V \in \mathbb{R}^{d \times d}$  are trainable parameters in transformation. The final representation of user is obtained by a linear combination of the two parts: temporal behaviors and social influence, shown as follows:

$$h_{\text{hybrid}} = \mathbf{W}_u(h_s || h_g), \quad (18)$$

where  $||$  denotes concatenation operation,  $\mathbf{W}_u \in \mathbb{R}^{d \times 2d}$  is a transformation matrix and  $h_{\text{hybrid}}$  is the final representation of the user. The prediction of next item is obtained through a softmax function.

$$Y = \text{Softmax}(h_{\text{hybrid}} \mathbf{I}^T), \quad (19)$$

where  $\mathbf{I} \in \mathbb{R}^{I \times d}$  is the embedding matrix of items,  $Y$  denotes the probabilities of items will be selected as the next one.

## 3.2 Model Optimization

In the training stage, we adopt the Bayesian Personalized Ranking (BPR) framework[25] for optimizing the recommendation task:

$$\mathcal{L}_{\text{BPR}} = \sum_{\hat{y}_{ui^+} \in D_t, \hat{y}_{ui^-} \in D_n} -\ln \sigma(\hat{y}_{ui^+} - \hat{y}_{ui^-}), \quad (20)$$

where  $D_t, D_n$  denote the training set and unobserved item set,  $\hat{y}_{ui^+}$  is the predicted probability of observed item  $i^+$  and  $\hat{y}_{ui^-}$  is the predicted probability of the sampled unobserved item  $i^-$ . To effectively learn from the temporal sequence, we consider the fitting of temporal point process by optimizing the maximum log-likelihood of observed items:

$$\begin{aligned} \mathcal{L}_{\mathcal{P}_n} &= -\sum_{i=1}^n \log(\lambda(t_i|H_{t_i})) + \int_0^T \lambda(t|H_t) dt \\ &= -\sum_{i=1}^n \left[ \log(\lambda(t_i|H_{t_i})) + \int_{t_i}^{t_{i+1}} \lambda(t|H_t) dt \right] \\ &= -\sum_{i=1}^n \log(\lambda_{n,i}^S) + \frac{1}{2} \sum_{i=1}^n \Delta t_i (\lambda_{n,i}^S + \lambda_{n,i+1}^S), \end{aligned} \quad (21)$$

where  $\Delta t_i$  is the time interval between observed items. It leverages trapezoidal rule, one of numerical approximations, to calculate the integral in Eq. (21). We finally adopt joint learning for optimizing both the loss of BPR and temporal point process, which is defined as Eq. (22):

$$\mathcal{L} = \mathcal{L}_{\text{BPR}} + \mathcal{L}_{\mathcal{P}_n} + \gamma ||\Theta||_2, \quad (22)$$

where  $\gamma$  and  $\Theta$  are regularization weights and the set of trainable parameters, respectively.

## 4 EXPERIMENTS

### 4.1 Experimental Setup

**4.1.1 Datasets.** To evaluate the performance of our proposed model and other baselines, we conduct experiments on three real-world datasets. We provide descriptive statistics of them in Table 1.

**Table 1: Basic Statistic of Datasets**

Dataset	Delicious	Yelp	Ciao
#users	1,867	74,982	6,596
#items	24,052	59,692	107,320
#Events	437,593	811,503	274,065
#Social Relations	15,328	2,577,679	114,893
Avg. events per user	234.38	10.82	41.55
Avg. relations per user	8.24	34.37	17.42

*Delicious.*<sup>1</sup> This dataset contains social networking and book-marking information from Delicious online social bookmarking system. It involves the sequence of tagging for each user with timestamps and relationships between users.

<sup>1</sup><https://grouplens.org/datasets/hetrec-2011/>

*Yelp*.<sup>2</sup> This dataset contains abundant users’ reviews for local businesses (it is also timestamped) on yelp platform. Here we treat each review as an interactive behavior with a business.

*Ciao*.<sup>3</sup> This dataset involves users’ rating records and social relations, e.g., trust relations with other users.

For data preprocessing, we follow the common previous works[11, 29] to convert all numeric ratings and reviews to implicit feedback with value 1. To avoid too large period, we choose first 10% data as a division for Yelp dataset to do experiments. To guarantee the quality of datasets, we filter out users that interacted less than 5 times and items appear less than 10 times for Yelp and 5 for the others. For all datasets, we use the last item of each sequence for testing and use the penultimate item for validation.

**4.1.2 Evaluation Metrics.** We adopt several common evaluation metrics to evaluate the recommendation performance, including Recall, Normalized Discount Cumulative Gain (NDCG) and Mean Reciprocal Rank (MRR):

- **Recall@k.** It means the percentage of groundtruth relevant items appear within the top- $k$  ranking list. Consider to next-item recommendation, it is equivalent to the hit ratio.
- **NDCG@k.** It is a standard ranking metric and reflects both the correlation and position for each recommended item. For next-item recommendation, it is formulated as  $NDCG = \frac{1}{\log_2(1+r_p)}$ , where  $r_p$  is the rank of the positive item.
- **MRR@k.** It takes the actual rank of target item into consideration and is calculated by the mean value of the reciprocal of target item’s actual rank. When the actual rank is out of  $k$ , the reciprocal rank is set to zero.

**4.1.3 Baseline Methods.** To evaluate the performance of STEN, we choose several comparative baselines, including state-of-the-art methods from both sequential and social recommendation.

- **BPR-MF**[24]: It utilizes matrix factorization and rating matrix to model latent factors of users and items with the loss of Bayesian pairwise ranking loss.
- **FPMC**[26]: This method captures users’ preference from their sequential behaviors based on Markov Chain.
- **SRGNN**[40]: This method treats user’s sequential behaviors as graph-structural data to capture features from complex transitions of items.
- **SASRec**[11]: This method adopts an attention mechanism to capture users’ sequential behaviors and makes prediction based on relatively few actions.
- **BERT4Rec**[29]: This method employs a deep bidirectional self-attention to model user behavior sequences and achieves strong performance.
- **THP**[45]: This method is a temporal point process model, which utilizes a transformer to capture the long-term dependency in the temporal sequence.
- **TiSASRec**[15]: TiSASRec is a time-sensitive method, which models both the absolute positions of items and the time intervals between them together.
- **SBPR**[43]: This method leverages social connections to enhance the performance of Bayesian Personalized Ranking.

- **DANSER**[39]: This method collaboratively learns representation of user-specific attention and context-aware attention in social information by dual graph attention networks.
- **Diffnet**[38]: Diffnet is a deep influence propagation model that stimulates how users are influenced by the recursive social diffusion process for social recommendation.

**4.1.4 Implementation Details & Parameter Settings.** In our experiments, we implement our model with Python 3.7 and Pytorch 1.7.0 and train it and other baselines on 4 Nvidia GTX 2080 GPU with 12G memory. In parameter settings, we use Adam[13] as the optimizer for all models. In addition, we also adopt mini-batch with batchsize of 256. The embedding size and learning rate for all methods are determined by grid search from {40, 60, 80, 100, 120, 140} and {0.001, 0.005, 0.01}, respectively. The parameters are randomly initialized to fit normal distribution  $N(0, 0.01)$ . To avoid overfitting and improve robustness, we add the dropout operation with dropout ratio tuning from {0.1, 0.2, ..., 0.9} and regularization term with coefficient tuning from {0.0001, 0.0005, 0.001, 0.005}. For a fair comparison, the results for all baselines are reported with the optimal hyper-parameter settings. For Diffnet, we remove the fusion component for both user and item to accommodate the absence of features. For THP, we transfer the task of event type prediction to item recommendation and remove the time prediction loss to improve its performance. For our model STEN, the max-order of the graph is set to 2. The sampling number of friends is set to 10 and the maximum sequence length depends on characteristics of datasets, we set {300, 30, 50} for three datasets, respectively.

## 4.2 Overall Performance

Table 2 shows the performance of our model and baselines over three datasets under the metrics of Recall@k, NDCG@k, MRR@k,  $k = \{10, 20\}$ . According to table 2, we can get the following observation.

The conventional recommendation method only takes the rating or interaction into consideration, without both the social relationship and the temporal information, which could explain why BPR-MF achieves poor performance.

For general sequential baselines, FPMC performs better than BPR-MF, which indicates the local sequential information is beneficial to sequential recommendation. We believe that the reason SR-GNN performs worst in the general sequential method is that it treats each session individually. SASRec and Bert4Rec both utilize attention mechanisms to capture sequential information. Bert4Rec overcomes SASRec on two datasets, which indicates the advantage of bi-directional networks. In addition, for time-aware sequential methods, e.g., THP utilizes the transformer-based neural Hawkes process to model the asynchronous temporal sequences. However, its performance is worse than other methods. It shows the limitation that it aims to obtain the event representation via the neural Hawkes process. Besides, we think the way to transfer THP into the recommendation task is too direct to be effective. TiSASRec extends SASRec by taking the time interval into consideration and achieves the second-highest results over three datasets, which demonstrates that time interval is an important factor in recommendation.

For social-based methods, SBPR is worse than the other two, which indicates GNN is a strong tool to model social relations. DANSER gets great improvement by utilizing the dual attention

<sup>2</sup><https://www.yelp.com/dataset>

<sup>3</sup><https://www.cse.msu.edu/~tangjili/datasetcode/truststudy.htm>



**Table 2: Comparison of STEN and baselines over three datasets. The boldface indicates the highest score and the underline indicates the second highest score. BPR-MF is conventional baseline. FPMC, SRGNN, SASRec and Bert4Rec are general sequential baselines. THP and TiSASRec are time-aware sequential baselines. SBPR, DANSER and Diffnet are social baselines.**

Dataset	Metrics	Conventional	Sequential				Time-aware		Social			STEN	Impr.
		BPR-MF	FPMC	SRGNN	SASRec	Bert4Rec	THP	TiSASRec	SBPR	DANSER	Diffnet		
Delicious	Recall@10	0.1935	0.2784	0.2370	0.2857	0.3125	0.2176	<u>0.3362</u>	0.2338	0.2969	0.3016	<b>0.3492*</b>	+3.87%
	Recall@20	0.2454	0.3188	0.2703	0.3133	0.3496	0.2676	<u>0.3695</u>	0.2661	0.3144	0.3238	<b>0.3731*</b>	+0.98%
	NDCG@10	0.1343	0.2317	0.1640	0.2145	0.2583	0.1454	<u>0.2702</u>	0.1687	0.1809	0.1963	<b>0.2833*</b>	+4.83%
	NDCG@20	0.1475	0.2419	0.1724	0.2215	0.2653	0.1562	<u>0.2785</u>	0.1792	0.1849	0.2031	<b>0.2885*</b>	+3.58%
	MRR@10	0.1161	0.2170	0.1411	0.1918	0.2353	0.1287	<u>0.2528</u>	0.1418	0.1535	0.1563	<b>0.2673*</b>	+5.74%
	MRR@20	0.1197	0.2197	0.1434	0.1937	0.2385	0.1291	<u>0.2547</u>	0.1446	0.1568	0.1646	<b>0.2692*</b>	+5.70%
Yelp	Recall@10	0.0353	0.0436	0.0367	0.0461	0.0452	0.0395	<u>0.0493</u>	0.0375	0.0451	0.0453	<b>0.0512*</b>	+3.80%
	Recall@20	0.0589	0.0739	0.0620	0.0758	0.0756	0.0649	<u>0.0815</u>	0.0624	0.0746	0.0763	<b>0.0846*</b>	+3.81 %
	NDCG@10	0.0177	0.0212	0.0183	0.0233	0.0226	0.0191	<u>0.0247</u>	0.0185	0.0227	0.0226	<b>0.0262*</b>	+6.29 %
	NDCG@20	0.0236	0.0288	0.0246	0.0308	0.0302	0.0255	<u>0.0327</u>	0.0247	0.0301	0.0303	<b>0.0344*</b>	+5.09 %
	MRR@10	0.0124	0.0145	0.0128	0.0165	0.0158	0.0131	<u>0.0172</u>	0.0128	0.0160	0.0161	<b>0.0185*</b>	+7.32 %
	MRR@20	0.0140	0.0166	0.0145	0.0185	0.0178	0.0148	<u>0.0194</u>	0.0145	0.0180	0.0182	<b>0.0207*</b>	+6.54 %
Ciao	Recall@10	0.0591	0.0593	0.0552	0.0624	0.0832	0.0676	<u>0.0960</u>	0.0705	0.0815	0.0897	<b>0.1019*</b>	+5.79 %
	Recall@20	0.0829	0.0912	0.0871	0.0972	0.1048	0.0976	0.1220	0.0984	0.1035	0.1232	<b>0.1294*</b>	+4.79 %
	NDCG@10	0.0304	0.0309	0.0301	0.0325	0.0523	0.0385	<u>0.0564</u>	0.0369	0.0455	0.0508	<b>0.0596*</b>	+5.37 %
	NDCG@20	0.0365	0.0388	0.0381	0.0413	0.0575	0.0421	<u>0.0645</u>	0.0439	0.0515	0.0597	<b>0.0685*</b>	+5.84 %
	MRR@10	0.0217	0.0223	0.0225	0.0235	0.0384	0.0249	<u>0.0427</u>	0.0266	0.0344	0.0389	<b>0.0462*</b>	+7.58 %
	MRR@20	0.0234	0.0244	0.0247	0.0259	0.0406	0.0149	<u>0.0438</u>	0.0286	0.0366	0.0418	<b>0.0479*</b>	+8.56 %

\* denotes the statistically significant improvements (i.e., one-tailed t-test with  $p \leq 0.05$ ) compared with the best baseline.

**Table 3: Ablation study on three datasets. (R@10 for Recall@10, N@10 for NDCG@10)**

Method	Delicious		Yelp		Ciao	
	R@10	N@10	R@10	N@10	R@10	N@10
w/o GE	0.3355	0.2722	0.0468	0.0229	0.0976	0.0582
w/o MT	0.3201	0.2646	0.0431	0.0215	0.0833	0.0490
w/o ST	0.3075	0.2529	0.0447	0.0221	0.0815	0.0482
w/o TC	0.3163	0.2617	0.0463	0.0232	0.0826	0.0484
Default	<b>0.3492</b>	<b>0.2833</b>	<b>0.0512</b>	<b>0.0262</b>	<b>0.1019</b>	<b>0.0596</b>

mechanism to capture the social information for both users and items. Diffnet, which uses the defusion graph structure to stimulate social influence propagation to improve the performance, has a similar performance to the former.

Finally, our proposed model STEN achieves the best performance on three datasets and all metrics. The reasons could be summarized as follows: (1) it introduces the temporal point process as the prototype to model both sequential order and time interval; (2) the designed structure mutually exciting attention network achieves the effect of directly modeling the influence of friends sequence on the target user and capture the temporal relationships between behavior events across users; (3) it utilizes a social heterogeneous graph to generate effective social node embedding.

### 4.3 Ablation Study

To justify the effectiveness of our components, we analyze the impact of utilization of social and temporal information via an ablation study over all datasets. Table 3 shows the performance of STEN and all variants. The variants and the analysis of experiment results are listed as follows:

(a) *Remove Social Graph Embedding Layer*. In our model STEN, social information is utilized by two components, the social heterogeneous graph network, and mutually exciting attention network. We replace the structural graph embedding with a trainable embedding lookup operation. The results show that it suffers a clear decrease compared to the default model, which denotes the structural view of social relationships is important to describe user and item representations.

(b) *Remove Mutual-Excitation Temporal Networks*. Here we remove the part of mutually exciting and directly utilize the social graph embedding to calculate the event conditional intensity. The result shows that it suffers a significant drop. The phenomenon that it performs worse than (a) could demonstrate the superiority of the direct paradigm. Without the mutually exciting temporal networks, the model degrades to a user-level model and suffers a lot of information loss.

(c) *Remove Self-Excitation Temporal Networks*. Here we replace the self-exciting module with average pooling. We can observe that it performs the worst among all variants. The result draws to the fact that the impact of the association between user’s own behaviors is more important than the impact of friend’s behaviors.

(d) *Remove Temporal Component*. To test the effectiveness of leveraging temporal information, here we remove the temporal component in STEN. Specifically, we remove the temporal term in Eq. (12) and the temporal loss function  $\mathcal{L}_{\mathcal{P}_n}$ . The results indicate that removing the temporal component leads to significant performance degradation. We believe when temporal information is lost, the model cannot capture the specific relationship between events.

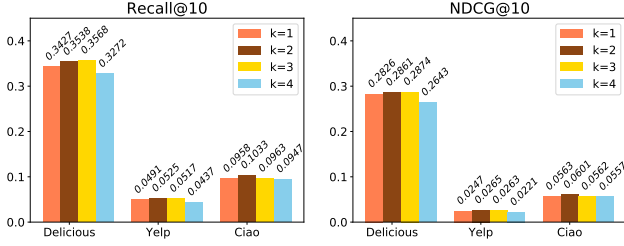


Figure 3: The impact of the order of neighbors

#### 4.4 Sensitivity Analysis

**4.4.1 Performance Comparison w.r.t. the Order of Neighbors.** As discussed in Section 3.1.1, we test the impact of a different order of neighbors in the social heterogeneous graph. Here we choose order  $k \in \{1, 2, 3, 4\}$  to do experiments and the higher-order social relation graph involves the information of the lower-order neighbors. Figure 3 illustrates the performance under the condition of a different order of neighbors. We can find that the performance is improved by the second-order graph. We analyze it for the reason that the second-order graph provides extra information on social connections and contributes to social sparsity. However, when the order becomes larger, the results change to worse. We consider that it is because the higher-order graph leads to graph convergence and noise, *i.e.*, the differences between users are narrowed.

**4.4.2 Performance Comparison w.r.t. the Window Size of Friend Sequence.** STEN extracts the event-level social influence through the mutually exciting temporal network. The larger window size brings richer social information and more complicated calculations. Hence, we explore the impact of the window size of friend behavior event sequence. As Figure 4 illustrates, with the increasing of the window size, the performance is first improved and then converges. When the window size is too short, the model only captures the current state of the friend, which contains a large deviation. The selection of window size depends on the average length of the user’s event sequence.

**4.4.3 Performance Comparison w.r.t. the Number of Sampled Friends.** To reduce the computation for users with complex relationships, STEN adopts a random sampling operation for their friends in mutually exciting temporal network. Figure 5 illustrates how the model’s performance varies with the number of sampled friends. The results show that the performance of the model improves first and then converges with the increase of the number of samples. When only one friend is sampled, the performance of the model decreases significantly. Compared with the rest, the result on Delicious shows it is insensitive to the number of sampled friends. We consider the reasons can be summed up into two points: 1) As Table 1 shows, the average relations is obviously less than the rest. 2) It has a smaller proportion of Time-Efficient Friends (TEF). The definition of TEF is that the sequence of behavior events has an intersection with the target user in time. The proportions of TEF over three datasets are shown in Table 4.

#### 4.5 Model Explainability

In this part, we conduct a case study on the Delicious dataset to show the explainability of our model. We visualize the attention weights

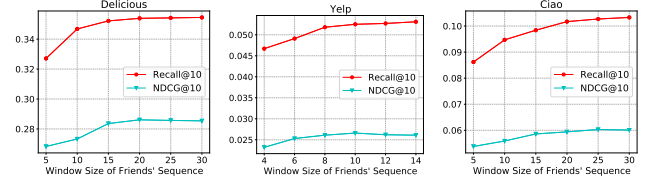


Figure 4: The impact of the window size of friend sequences.

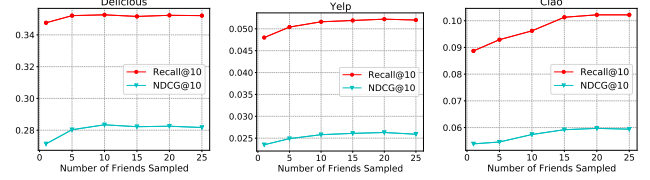


Figure 5: The impact of the number of sampled friends.

Table 4: The proportions of time-efficient friends

Dataset	Delicious	Yelp	Ciao
TEF(%)	59.16	97.75	94.93

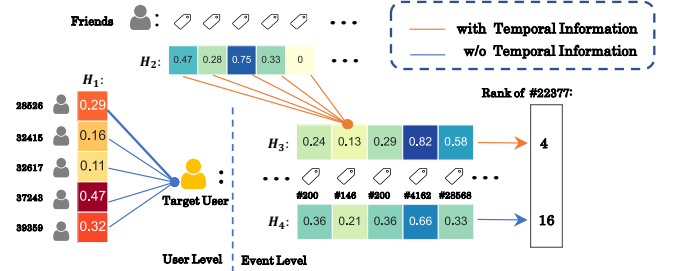


Figure 6: Visualization of attention weights in STEN.  $H_1$  denotes the impact of friends to target user.  $H_2$  denotes the impact of friends events on target user event.  $H_3, H_4$  denote the impact of target user behavior events on themselves, where  $H_3$  involves the temporal information, but  $H_4$  does not.

of different modules in Figure 6. The value in  $H_1$ , which is obtained from the graph network component, indicates the impact of friends on to target user. It shows the coarse-grained relationships at the user level, while  $H_2, H_3, H_4$  describe the fine-grained relationships at the event level.  $H_2$  is obtained from mutually exciting temporal network, which indicates the impact of a friend’s events on event #146. The value 0 in  $H_2$  means that the event happens after event #146, which obeys the rule that no future information is leveraged.  $H_3, H_4$  indicate the impact of target user behavior events on themselves. The difference is that  $H_3$  involves temporal information and event-level social relation, while  $H_4$  only has user-level social information. Besides, we use these two parts to make recommendations separately and show the predicted position of the ground truth event. It is clear that the two parts have different focuses on the event sequence and affect the performance of the model.



## 4.6 Model Complexity

We evaluate the temporal and space complexity of STEN by recording the inference time and the model parameters. We compare STEN with the best-performing baselines from three categories mentioned in section 4.2 and results are reported in Table 5. As online recommendation services usually have high requirements on latency, the computational cost during inference is more important than that of the training stage. Compared with the baselines, the increased inference time of STEN is 11% on average, which denotes the efficiency of our model. Besides, we use the number of model parameters to measure the space complexity of models. Despite STEN has performed fine-grained modeling at the event level, it gets the 10% increase on average, within an acceptable range. Compared with the embeddings of both users and items, the parameters in heterogeneous graph component and temporal excitation networks only occupy a small proportion.

**Table 5: Comparison of model complexity on Ciao dataset.**

Method	Inference Time	INCR.	Parameters	INCR.
Bert4Rec	~3.82s	+6.80%	~3.65×10 <sup>6</sup>	+7.74%
Diffnet	~3.35s	+21.79%	~3.59×10 <sup>6</sup>	+9.53%
TiSASRec	~3.91s	+4.45%	~3.48×10 <sup>6</sup>	+12.74%
STEN	~4.08s	-	~3.93×10 <sup>6</sup>	-

## 5 RELATED WORK

In this section, we briefly review the related research work from three aspects: social recommendation, general sequential recommendation, and time-aware sequential recommendation.

**Social Recommendation.** Leveraging social information has played an essential role in alleviating the problem of data sparsity and enhancing recommendation. Kautz et al. [12] import social networks into traditional method collaborative filtering for item recommendation. Ma et al. [17] propose SoRec that incorporates trust network into user tastes analysis through probabilistic matrix factorization. In addition, factor-based random walk is combined with a coupled latent factor model in[41] to deal with both social connections and interest interactions. Besides, Zhao et al. [43] extend the framework of social networks with collaborative filtering to the one-class recommendation. With the development of neural networks, some research works[5, 36] utilize deep neural networks to extract information of social networks and learn the complex latent representation. ARSE[30] combines attention mechanism with social LSTM to leverage the temporal information to enhance social recommendation. GraphRec[7] utilizes GNNs to model user-user and user-item graphs and heterogeneous strengths in social recommendation. To better simulate the recursive diffusion in social networks, Diffnet[38] is designed with an influence propagation structure to model users’ latent embedding. To learn latent features from both neighbors and related items, a dual graph attention network model named DANSER[39] is proposed and it solves the problem of diversity of neighbor’s influence. DGR[28] utilizes RNNs to model users’ dynamic interests and generates the user representation firstly and then combines it with their social relationship through GNNs.

**General Sequential Recommendation.** Sequential recommendation has attracted a lot of attention from both academia and industry. The key of sequential recommendation is to extract the hidden information from the user behavior sequence. Rendle et al. [26] propose FPMC, which imports Markov Chain (MC) to matrix factorization for user state transition. In addition, RNN and its variants, *e.g.*, Long Short-Term Memory (LSTM) [10] and Gated Recurrent Unit (GRU) [3] are used to model the sequential behaviors of user for their latent features in many related studies[9, 37, 42]. However, RNN-based models usually suffer from low computational efficiency. CNN-based model Caser[32] is proposed to use a bi-direction convolution filter to learn sequential patterns. To improve the ability of modeling long sequence, attention-based method Transformer[33] and BERT[4] are utilized by SASRec[11] and BERT4Rec [29] to capture sequential information, respectively. Wu et al. propose SRGNN[40] that utilizes Gated graph neural network[16] to capture structure information from sequential data.

**Time-aware Sequential Recommendation.** To take the time interval of user actions into consideration, there also develops many time-sensitive recommendation methods. Time-LSTM[44] captures the user’s long short-term dynamic interests through the specific time gates. Based on SASRec[11], TiSASRec[15] take the time interval between different actions into consideration and model different time intervals as relations between any two interactions. Besides, another way for the time-sensitive method is to formulate the sequence as a temporal point process. LSHP[2] models the long-term and short-term dependency between users’ online interactive behaviors through a Hawkes Process. With the development of neural point process, *e.g.*, neural Hawkes Process[21], CTRec[1] utilizes a convolutional LSTM and attention mechanism to capture short-term and long-term interests, respectively, based on a demand-aware Hawkes Process. However, RNN-based point process models have limitations in both capturing long-term dependency and computational efficiency. Hawkes process is combined with self-attention mechanism in[45] to alleviate the limitations. The above time-aware studies leverage the temporal information to model the behavior sequence of individual users. By contrast, our model STEN aims to leverage the temporal information not only to model the user’s dynamic interests but also to capture the relation among users through their temporal mutual dependency.

## 6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel time-aware sequential recommendation model STEN, to model the fine-grained temporal behavior sequences of users with social information. STEN establishes a social heterogeneous graph to learn the node representations of users and items. In order to model the time-aware behavior sequence, we introduce two classical temporal point processes as prototypes to enhance STEN. STEN models the impact of friends behavior sequences and user’s behavior sequence in a direct paradigm by mutually exciting temporal network and self-exciting temporal network. Compared with the previous methods, STEN could model the temporal relationships between cross-user events. As a future plan, we will improve the current model to be type-aware to better model the influence of different types of events and to expand its applicable scenario.

## REFERENCES

- [1] Ting Bai, Lixin Zou, Wayne Xin Zhao, Pan Du, Weidong Liu, Jian-Yun Nie, and Ji-Rong Wen. 2019. CTRec: A Long-Short Demands Evolution Model for Continuous-Time Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 675–684.
- [2] 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*. 873–882.
- [3] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.
- [4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [5] Wenqi Fan, Qing Li, and Min Cheng. 2018. Deep modeling of social relations for recommendation. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*. AAAI press, 8075–8076.
- [6] Wenqi Fan, Yao Ma, Qing Li, Yuan He, and Dawei Yin. 2019. Graph Neural Networks for Social Recommendation. (2019).
- [7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *The World Wide Web Conference*. 417–426.
- [8] ALAN G. HAWKES. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (04 1971), 83–90. <https://doi.org/10.1093/biomet/58.1.83> arXiv:<https://academic.oup.com/biomet/article-pdf/58/1/83/6026258/58-1-83.pdf>
- [9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.
- [10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
- [11] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.
- [12] Henry Kautz, Bart Selman, and Mehul Shah. 1997. Referral Web: combining social networks and collaborative filtering. *Commun. ACM* 40, 3 (1997), 63–65.
- [13] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [14] 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*. ACM, 1419–1428.
- [15] Jiacheng Li, Yujie Wang, and Julian McAuley. 2020. Time Interval Aware Self-Attention for Sequential Recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 322–330.
- [16] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. 2015. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493* (2015).
- [17] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. 931–940.
- [18] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*.
- [19] Peter V. Marsden and Noah E. Friedkin. 1994. Network Studies of Social Influence. *Sociological Methods & Research* 22, 1 (1994), 127–151.
- [20] McPherson, Miller, Smith-Lovin, Lynn, Cook, and M. James. 2001. BIRDS OF A FEATHER: Homophily in Social Networks. *Annual Review of Sociology* (2001).
- [21] Hongyuan Mei and Jason M Eisner. 2017. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Curran Associates, Inc., 6754–6764. <http://papers.nips.cc/paper/7252-the-neural-hawkes-process-a-neurally-self-modulating-multivariate-point-process.pdf>
- [22] Nan Mu, Daren Zha, Yuanhe He, and Zhihao Tang. 2019. Graph attention networks for neural social recommendation. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 1320–1327.
- [23] Takahiro Omi, Kazuyuki Aihara, et al. 2019. Fully neural network based model for general temporal point processes. In *Advances in Neural Information Processing Systems*. 2122–2132.
- [24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and ST Lars. 2009. Bpr: Bayesian personalized ranking from implicit feedback. *UAI 09. Arlington, Virginia, United States* (2009), 452–461.
- [25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).
- [26] 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*. ACM, 811–820.
- [27] P. J. Resnick and Hal R Varian. 1997. Recommender systems. *Communications of the Acm* (1997).
- [28] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *Proceedings of the 28th ACM International Conference on Web Search and Data Mining*. 555–563.
- [29] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1441–1450.
- [30] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive recurrent social recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 185–194.
- [31] Jiliang Tang, Huiji Gao, Xia Hu, and Huan Liu. 2013. Exploiting homophily effect for trust prediction. In *Proceedings of the sixth ACM international conference on Web search and data mining*.
- [32] Jiayi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *the Eleventh ACM International Conference*.
- [33] 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 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Curran Associates, Inc., 5998–6008.
- [34] Menghan Wang, Xiaolin Zheng, Yang Yang, and Kun Zhang. 2017. Collaborative Filtering with Social Exposure: A Modular Approach to Social Recommendation. (2017).
- [35] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. 2019. Heterogeneous graph attention network. In *The World Wide Web Conference*. 2022–2032.
- [36] Yufei Wen, Lei Guo, Zhumin Chen, and Jun Ma. 2018. Network embedding based recommendation method in social networks. In *Companion Proceedings of the The Web Conference 2018*. 11–12.
- [37] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. 495–503.
- [38] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*. 235–244.
- [39] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.
- [40] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 346–353.
- [41] Shuang-Hong Yang, Bo Long, Alex Smola, Narayanan Sadagopan, Zhaohui Zheng, and Hongyuan Zha. 2011. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web*. 537–546.
- [42] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. 729–732.
- [43] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 261–270.
- [44] 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 *IJCAI*, Vol. 17. 3602–3608.
- [45] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer hawkes process. In *International Conference on Machine Learning*. PMLR, 11692–11702.