Highlights

Event-based Incremental Recommendation via Factors Mixed Hawkes Process

Zhihong Cui, Xiangguo Sun, Li Pan, Shijun Liu, Guandong Xu

- We distinguish intrinsic intensity, external intensity, and historical intensity to explore how each event happens.

- We propose novel Factors Mixed Hawkes Process (FMHP) for the event-based incremental recommendation.

- FMHP can incrementally update the event formation when new events occur.

- We have conducted expensive experiments on four large real-world data sets.

Event-based Incremental Recommendation via Factors Mixed Hawkes Process

Zhihong Cui[a], Xiangguo Sun[b], Li Pan[a], Shijun Liu[a,c,*], Guandong Xu[d,*]

[a]*Shandong University, Jinan, Shandong, China*
[b]*The Chinese University of Hong Kong, New Territories, Hong Kong, China*
[c]*Quan Cheng Laboratory, Jinan, Shandong, China*
[d]*The University of Technology Sydney, Sydney, NSW, Australia*

Abstract

Incremental recommendation systems have garnered significant research interest since they ideally adapt to users' ongoing events (such as clicking, browsing, and reviewing) and recommend items without retaining the model. Many methods have tracked the event generation sequences for incremental recommendation. However, most existing models treat the event as a static snapshot or a black box, ignoring the underlying factors that may trigger the event generation. Underlying such inner factors can help RS reasonably and foreseeingly evaluate the potential items for the user next time. Along this vein, we propose the Factors Mixed Hawkes Process (FMHP) for event-based incremental recommendations. First, we extend each event to a notion of factor-driven event sequence. Next, we consider three factors that may influence the occurrence of an event: intrinsic intensity, external intensity, and historical intensity. An intrinsic intensity function, multi-type temporal attention, and a hybrid time decay function are incorporated in FMHP to evaluate the intrinsic, external, and historical intensity, respectively. In addition, an incremental updating strategy is implemented in FMHP, continuously updating event intensity as new events occur. We conduct extensive experiments on four public datasets (e.g., Amazon Beauty, LastFM, Movielens, and Amazon Book). Compared with state-of-the-art incremental recommendation methods, our proposed FMHP model achieves superior performance, with up to 9.77%, 9.35%, 9.32%, 10.10% *w.r.t* HR, 8.86%, 10.26%, 9.81%, 9.38% *w.r.t* NDCG, and 9.64%, 9.32%, 8.97%,9.85% *w.r.t* Recall in Beauty, LastFM, MovieLens, and Book, respectively. Besides, the case study shows that the three factors in our proposed FMHP method play a vital role in triggering event generation.

Keywords: incremental recommendation, Hawkes process, events, dynamic graphs

1. Introduction

Recommendation System (RS) [36, 2] aims to recommend an item for the target user based on his historical interactions, which plays a crucial role in helping users engage with shopping platforms. Generally, user events [21] (such as clicking, browsing, and reviewing) occur continuously over time, making the RS incremental. It means RS should ideally adapt to the user's these events, incrementally modifying the model without retaining it for a recommendation. For example, if a user clicks Mac at time t_1 and then clicks AirPods at t_2, the RS could incrementally update the model to recommend items at t_2 instead of retraining the model using all events in t_1 and t_2. Generally, for incremental recommendation [21, 29], RS needs to understand the underlying generation reasons of the user's every event, allowing it reasonably foresees the continuous transmission process of events and recommend potential items for the user next time. In the literature on incremental recommendations, some methods [5, 44] treat each event as a static graph snapshot within a specific time and incrementally update each event for the recommendation. However, these methods aim to model the observed event as a whole graph, ignoring the emergence of every event in the graph. Recently, some methods [3, 25, 21] have sought to track the sequence of neighborhood formation for each user event. For example, [3, 25] model the event generation based on historical events with multi-aspect embedding. [25] treats each event as a micro-dynamic result based on historical events, constrained by the macro-network evolution. However, these methods treat the observed events as a black box and do not explain the underlying reasons triggering the generation of each event.

To explain the underlying reasons triggering the generation of each event, we distinguish three factors: intrinsic intensity, external intensity, and historical intensity. The intrinsic intensity [45, 34] refers to users' inherent preference, which triggers the user to click an item spontaneously. After analysis of the real-world data, we find that each user has a specific taste for each item at a given time. For example, from the user's historical click sequence, we find that a user may frequently click on mobile phones but rarely browse clothes within a short time window. Thus, capturing the user's intrinsic intensity can help the recommendation system effectively locate the user's requirements. The external intensity [24, 32] triggers an event due to the semantics influence (such as brand, genre, or category) of

*Corresponding author
Email addresses: czh@mail.sdu.edu.cn (Zhihong Cui), xgsun@se.cuhk.edu.hk (Xiangguo Sun), panli@sdu.edu.cn (Li Pan), lsj@sdu.edu.cn (Shijun Liu), Guandong.xu@uts.edu.au (Guandong Xu)

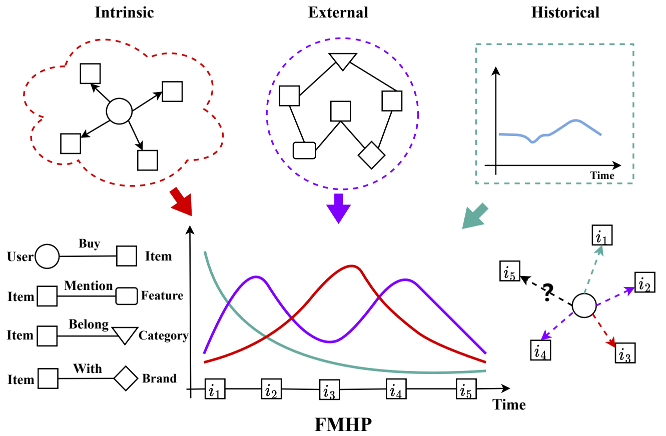


Figure 1: A toy example shows how three factors jointly trigger events over time. Each color represents the intensity change of a different factor. However, the three factors contribute differently to triggering events over time. This example shows that u clicked items i_1, i_2, i_3, and i_4 sequentially, whose dominant underlying reasons are historical, external, intrinsic, and external intensity. Jointly considering the changes of three factors' intensities, the proposed FMHP model calculates the probability of the user clicking i_5 and figures out its dominant underlying reasons.

the historical events. As mentioned in [24, 32], the semantics of events play essential guidance in recommendations. Inspired by this, we suspect each user may be concerned about the different semantics of an event. For example, if a user clicks an iPhone, which semantic of iPhone does he concern about more? Color? Brand? Price? The extra concern about the different semantics of the iPhone may lead to totally different recommendations. Thus, the user's external intensity can help the recommendation system capture items with more similar semantics to the user's concern. Since the events are continuously occurring, the historical intensity of each event is also an essential factor in triggering the event generation. Generally, the historical intensity [18, 31] measures the temporal influence of different historical events on the current event. For example, if a user clicked the Airpods a day ago and the Mac one year ago. In this example, a recent happened event (Airpods) may have more influence than an event (Mac) that happened a long time ago. To show how these three factors trigger events over time, we visualize them in Figure.1.

To this end, we propose a novel Factors Mixed Hawkes Process method for incremental recommendation (FMHP), which explores underlying factors that trigger event generation and incrementally updates the event embedding when new events occur. FMHP first extends each event to a notion of the factor-driven event sequence by plugging its various historical events until time t. Thus, each event is regarded as an event sequence in a finite time interval. Since the Hawkes process [3, 41] can well capture the arrival rate of the current event driven by historical events, particularly the occurrence probability of events at a given time, we adapt it here to model the generation process of each event. Next, to measure the intrinsic, external, and historical intensity of events, we introduce the intrinsic intensity function, multi-type temporal attention, and hybrid time decay function. These three intensities functions are incorporated into the Hawkes process to calculate the event intensity jointly at a given time. In addition, we have implemented an incremental updating algorithm to explain the strategy for updating the event intensity, allowing FMHP to update the event intensity incrementally when new events occur.

The main contributions are summarized as follows:

- We consider intrinsic, external, and historical intensity to explore how each event generates, which provides a more comprehensive analysis of events formation in dynamic graphs.
- We propose novel Factors Mixed Hawkes Process (FMHP) for the event-based incremental recommendation, which models the underlying factors that trigger the generation of each event and incrementally updates the event embedding when new events occur.
- We have conducted expensive experiments on four real-world datasets. Our experimental results demonstrate that (1) the three factors are vital in triggering event generation. (2) the superior recommendation performance of our FMHP method compared to seven state-of-the-art baselines.

The remainder of the paper is organized as follows. First, in section 2, we present some definitions and declare our problem. Next, we introduce the proposed FMHP method in Section 3. Next, various experimental results and analyses are reported in Section 4, followed by the related work in Section 5. Finally, we provide some conclusions and future work in Section 6.

2. Preliminaries

In this section, we formally define the dynamic events graph, factor-driven temporal sequence, Hawkes process, and the problem of factor-driven event sequence embedding for the event-based incremental recommendation. We list all notations in Table1.

Definition 1. (**Dynamic Events Graph**) A dynamic events graph [20, 38] can be denoted as $G = (E, R, T)$, where E is the edge set, R is the event type set, and T is the timestamps. Specifically, we define user' each event as $e = (u, i, r, t) \in E$, which indicates user $u \in U$ adopt an item $i \in I$ at time t, the event type is r.

Definition 2. (**Factor-Driven Event Sequence**) For user' each event $e = (u, i, r, t) \in E$, its factor-driven event sequence S [19, 27] is defined as a combination of the historical events $S_{u,<t}$ and the target node $S_{i,<t}$. Take the $S_{u,<t}$ as an example, starting from source node u, it contains all historical events [7, 12] from time t_1 to time t, denoted as $S_{u,<t} = \{(p_1, r_{p_1}, t_1), ..., (p_N, r_{p_N}, t), p \in E, r_p \in R, t_p \in T\}$. Similarly, the $S_{i,<t}$ is defined as $S_{i,<t} = \{(q_1, r_{q_1}, t_1), ..., (q_N, r_{q_N}, t), q \in E, r_q \in R, t_q \in T\}$. From the above definition, we find that each historical event p in $S_{u,<t}$ or q in $S_{i,<t}$ triggers event e at time t from three factors: the event itself (intrinsic intensity), the semantic of the event (external intensity) and the occurring time of the event (historical intensity).

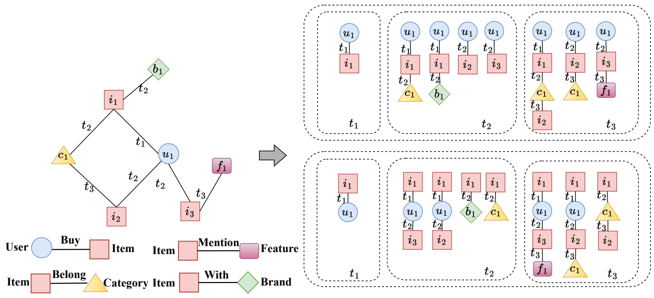
Table 1: Notations in our FMHP method

Notation	Description
U, u	the user set, a user or the source node of an event
I, i	the item set, an item or the target node of an event
R, r	the event type set, a type of an event
E, e	the event set, an event $e = \{u, i, r, t\}$
T, t	the timestamp of the event
p	the historical event of source node u
t_p	the timestamp of the historical event p
r_p	the event type of historical event p
h_p	the embedding of historical event p
$S_{u,<t}$	the factor-driven event sequence of source node u until time t
$S_{i,<t}$	the factor-driven event sequence of target node i until time t
$\hat{\lambda}(e)$	the intensity of an event e (formula 2)
$\mu(i, j)$	the intrinsic intensity of the event $e = (u, i, r, t)$
$\alpha_{r_{p,u}}(p, u)$	the type-wise attention function (formula (4)), calculating the semantics of external intensity of historical events p.
$\tilde{\alpha}_{r_{p,u}}(p, u)$	the type-wise weight between historical event p and source node i (formula (5))
δ_1	the temporal attention function (formula(7)), averaging the temporal effect of external intensity on all historical events.
$\kappa(t - t_p)$	the historical intensity function between historical event p and source node i (formula (12))
$N_{p,t}$	the occurring number of p at time t
t_{true_p}	the true time of historical event p
t_{pseudo_p}	the pseudo time of historical event p (formula(11))
$\tilde{h}_u$	the embedding aggregation of node u's historical event h_p with type-wise weight $\tilde{\alpha}_{r_{p,i}}$ in formula (5)
o_u	the final embedding of the user u (formula(13))
o_i	the final embedding of the item i
$\hat{y}_{u,i}$	the probability of user u interacting with item i

Example 1. As shown in Figure.2, we generate the factor-driven event sequence for the event (u_1, i_1, buy) at time t_1, t_2 and t_3. At each time, its event sequence is the combination of $S_{u_1,<t}$ and $S_{i_1,<t}$. Each event in $S_{u_1,<t}$ or $S_{i_1,<t}$ has different semantics, which indicates its external intensity. Each event occurs at a time, indicating its historical influence. For example, at time t_2, starting from u_1, we get its event sequence (u_1, i_1, buy, t_1) and (i_1, c_1, belong, t_2). Their event types are "buy" and "belong", respectively, representing different semantics. Besides, they also happen at t_1 and t_2, respectively, indicating different historical intensities.

Definition 3. (**Hawkes Process**) Hawkes process [3, 41] is a typical temporal point process for modeling network dynamics, as it considers the temporal decay effect of historical events when measuring the probability of an event. The conditional intensity function for the Hawkes process, denoted as $\lambda(t)$, is defined as follows:

$$\lambda(t) = \mu(t) + \int_{-\infty}^{t} \kappa(t - s) dN(s) \tag{1}$$



(a) Dynamic events graph

(b) Generating factor-driven event sequence for event (u_1, i_1) at time t_1, t_2, and t_3, respectively

Figure 2: A toy example shows how to generate a factor-driven event sequence for the event (u_1, i_1, buy) at time t_1, t_2 and t_3, respectively. At each time t, the sequence starts at either the source node u_1 or the target node i_1 and consists of historical events from time t_1 to the current time t.

where $\mu(t)$ represents the base rate, which describes the arrival rate of spontaneous events. $\kappa(t-s)$ models the time decay effect of past events on current events. $N(s)$ is the number of events up until time t.

Problem 1. (**Factor-Driven Event Sequence Embedding for Event-based Incremental Recommendation**) For each temporal event (u, i, r, t) and its factor-driven event sequence $S_{u,<T}$ and $S_{i,<T}$, we aim to learn its intensity from three factors: the intrinsic intensity, external intensity, and historical intensity and incrementally update the event intensity to predict the probability of the event (u, i) at time t. At a given time t, the top-k items are recommended for the user based on the probability.

3. Methodology

3.1. Overview

We propose a novel Factors Mixed Hawkes Process (FMHP) for event-based incremental recommendations (as shown in Figure.3). This method models the underlying factors that trigger the generation of each event and incrementally updates the event as new events occur. First, we extend the event to a notion of the factor-driven event sequence by plugging various historical events up until time t, which is demonstrated in section 2. Since the Hawkes process can well capture the occurring probability of the current event driven by historical events at a given time, we adapt it here to model the generation process of each event. Then, FMHP distinguishes three factors that may trigger the occurrence of each event: intrinsic intensity, external intensity, and historical intensity. An intrinsic intensity function, multi-type temporal attention, and a hybrid time decay function are incorporated into FMHP to measure the three factors separately. In addition, FMHP uses an incremental updating strategy to calculate the event intensity incrementally by feeding the event sequence into the model time by time. Finally, for the recommendation task, we generate user and item embeddings at time t by multiplying and adding the intensity of historical events with corresponding embedding one by one. We compute the predicted probability using a multi-layer perceptron (MLP) [17, 42] and select top-k items as recommendations for the user based on this probability.

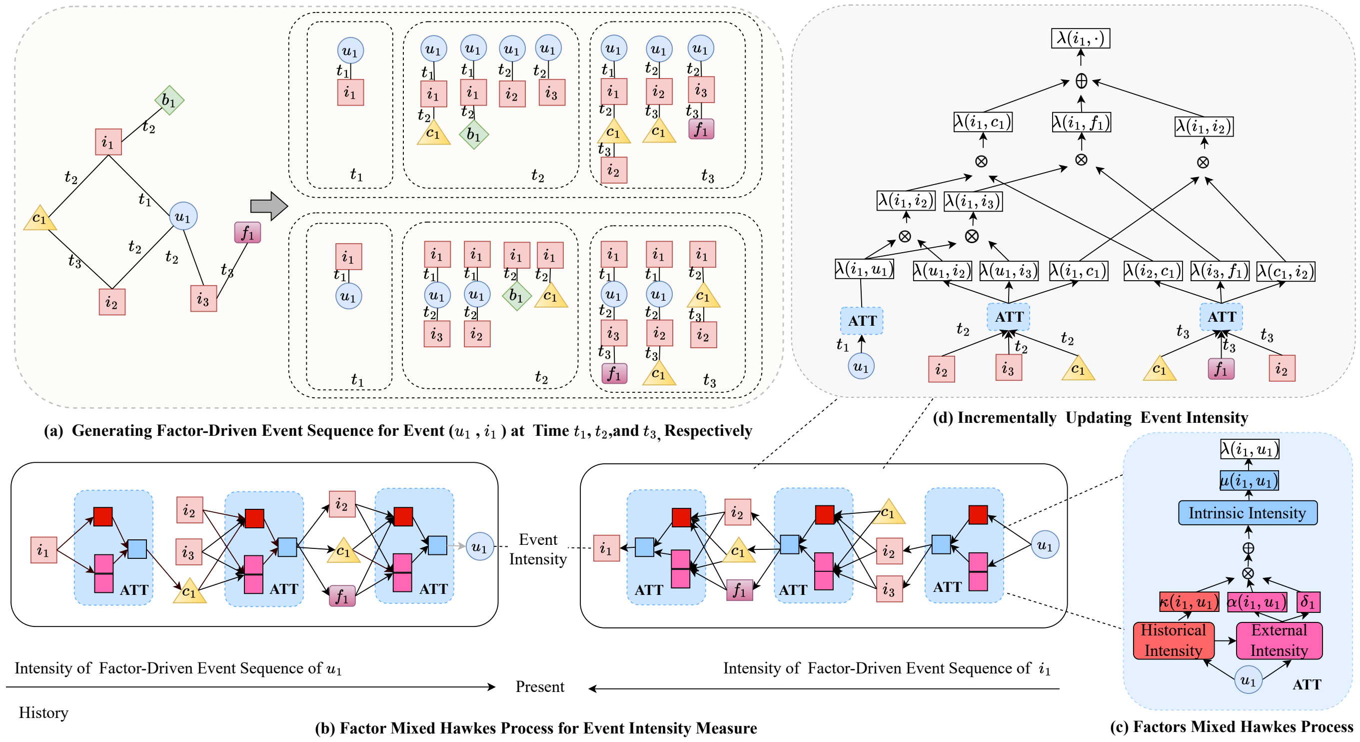


Figure 3: The overall architecture of FMHP for the event (u_1, i_1, buy, t_3). (a) Generating factor-driven event sequence of u_1 and i_1 for Event (u_1, i_1) at time t_1, t_2, and t_3, respectively. (b) Calculating the intensity of (u_1, i_1, buy, t_3) based on FMHP based on the factor-driven event sequence of source node u_1 and the target node i_1. (c) The factor mixed Hawkes process evaluates the intensity between historical node u_1 and target node i_1 at t_1 from three factors. (d) Incrementally updating the event intensity based on historical events of i_1.

3.2. *Event Intensity Measure Via Mixed Factors Hawkes Process*

Let $e = (u, i, r, t)$ be an event that happens at time t, where u is the source entity, which usually means the initiator of this event, i is the target entity that the user might click, r is the event type, and t is the event time. Then, as mentioned above, an event is triggered by the intrinsic intensity between the source node u and the target node i, the external influence related to the type-wise semantics of historical events in factor-driven event sequence, and the historical intensity related to the time decay effect of historical events in the factor-driven event sequence. To this end, we propose the Factor Mixed Hawkes Process to incrementally measure the event intensity of $e = (u, i, r, t)$:

$$\begin{aligned}\tilde{\lambda}(e_{u,i,r,t}) = \; & \underbrace{\mu(u,i)}_{\text{intrinsic intensity}} \\ + \; & \underbrace{\delta_1 \Sigma_{t=1}^{t} \Sigma_{p \in S_{u,<t}} \alpha_{r_{p,u}}(p,u) \kappa(t-t_p)}_{\text{Intensity of historical events on source node u}} \\ + \; & \underbrace{\delta_2 \Sigma_{t=1}^{t} \Sigma_{q \in S_{i,<t}} \alpha_{r_{q,i}}(q,i) \kappa(t-t_p)}_{\text{Intensity of historical events on target node i}}\end{aligned} \tag{2}$$

where $\mu(u, i)$ represents the intrinsic intensity; $\alpha_r(\cdot, \cdot)$ is the external influence, which measures type-wise semantic of the historical events; and $k(t - t_p)$ measures the influence of time decay of the historical events. We elaborate on these three factors by each in the following.

3.2.1. *Modelling Intrinsic Intensity*

The intrinsic intensity measures the event semantics between the source node u and the target node i. A key issue is to quantify the semantic intensity between u and i. Generally, given an event $e = (u, i, r, t)$, its semantic intensity is related to not only the u and i but also event type r. Thus, we define the intrinsic intensity $\mu(u, i)$ of an event as follows:

$$\mu(u, i) = -\sigma(f(h_u W_{r_u} - h_i W_{r_i}) W_{r_{u,i}} + b_{r_{u,i}}) \tag{3}$$

where $h_u \in \mathbb{R}^d$ and $h_i \in \mathbb{R}^d$ are the embedding of node u and i, d is the node embedding dimension. $W_{r_u} \in \mathbb{R}^d$ and $W_{r_i} \in \mathbb{R}^d$ denote the type projection matrix. $f(\cdot)$ is the element-level operation to measure the similarity of u and i. $W_{r_{u,i}}$ and $b_{r_{u,i}}$ are the type projection and bias of the event, $\sigma(\cdot)$ is the non-linear activate function.

3.2.2. *Modelling External Intensity*

Different types of events indicate different semantics. Besides, the semantics of events may change over time. Considering these, we define the multi-type attention to calculate the external intensity. Take the source node u and its historical event p as an example. The multi-type attention contains two attentions: type-wise attention $\alpha_{r_{p,u}}(p, u)$ [22, 39] and temporal attention δ_1 [43]. The type-wise attention $\alpha_{r_{p,u}}(p, u)$ is defined as:

$$\alpha_{r_{p,u}}(p, u) = \frac{exp(\tilde{\alpha}_{r_{p,u}}(p, u))}{\Sigma_{p \in S_{u,<t}} exp(\tilde{\alpha}_{r_{p,u}}(p, u))} \tag{4}$$

$$\tilde{\alpha}_{r_{p,u}}(p,u) = \sigma(\kappa(t-t_p)[h_p W_{r_p} \oplus h_u W_{r_u}]) \quad (5)$$

where $W_{r_p} \in \mathbb{R}^d$ and $W_{r_u} \in \mathbb{R}^d$ denote the type weight of historical event p and the source node u. $\oplus$ is the concentric operation. Since the semantics between the source node u and historical event p can differ over time, we also incorporate the time decay effect $\kappa(t-t_p)$. t and t_p are the hybrid time of the source node and the historical node, which will be introduced in section 3.2.3. $\tilde{\alpha}_{r_{p,u}}(p,u) \in \mathbb{R}^{2d}$ is the type-wise weight that measures the semantic importance of historical event type r_p on the source node type r_u.

Furthermore, to balance the time evolution effect of all historical event p on target event $e = (u, i, r, t)$, we further define temporal attention δ_1:

$$\tilde{\delta}_u = s(\kappa(\overline{t-t_p})\tilde{h}_u),\ \tilde{\delta}_i = s(\kappa(\overline{t-t_q})\tilde{h}_i) \quad (6)$$

$$\delta_1 = \frac{exp(\tilde{\delta}_u)}{exp(\tilde{\delta}_u) + exp(\tilde{\delta}_i)} \quad (7)$$

where $\tilde{\delta}_u$ and $\tilde{\delta}_i$ are the generalized temporal attention of the u's and i's historical events on the current event. $s(\cdot)$ is a single-layer neural network. $\tilde{h}_u$ and $\tilde{h}_i$ are the aggregation of node u's and i's historical nodes with type-wise weight, respectively:

$$\tilde{h}_u = \sigma(\sum_{p \in S_{u,<t}} \tilde{\alpha}_{r_{p,u}}(p,u) h_p) \quad (8)$$

where $\tilde{\alpha}_{r_{p,u}}$is the type-wise weight calculated by formula (5). h_p is the embedding of p. $\overline{t-t_p}$ is the averaged time interval of historical nodes, defined as:

$$\overline{t-t_p} = \frac{1}{|S_{u,<t}|} \sum_{p \in S_{u,<t}} (t-t_p). \quad (9)$$

3.2.3. Modelling Historical Intensity

The historical intensity $\kappa(\cdot)$ measures the time effect of the historical node on the event, which should decay with time. However, after analyzing the existing data, we find that each node has a unique occurrence ratio at different times (As shown in Figure.2). Considering this, we define a hybrid time for each event:

$$t_p = t_{true_p} - t_{pseudo_p} \quad (10)$$

where t_{true_p} and t_{pseudo_p} are the true time and the pseudo of p, respectively. Specifically, the true time t_{true_p} is the time interval between event p and node u. The pseudo time t_{pseudo_p} is its occurrence frequency accumulated until time t.

$$t_{pseudo_p} = \frac{1}{T} \sum_{t=1}^{T} \sum_{p \in S_{u,<t}} \frac{N_{p,t}}{N_t} \quad (11)$$

where $N_{p,t}$ is the occurrence number of event p at time t. N_t is the number of p in the t-th temporal events set. To this end, the hybrid time decay function $\kappa(t-t_p)$ is defined as follows:

$$\kappa(t-t_p) = exp(-\sigma(t-t_p)) \quad (12)$$

As we can see from the formula (10), the more frequently a historical node is sampled, the greater its impact on the target event. Additionally, the shorter the time interval of a historical node, the greater its impact on the target event.

3.3. Incremental Updating Strategy for New Events

We now describe the strategy of incremental updating [10] for new events. When a new event $x = (w, v, r, t+1)$ occurs at time t+1, one hypothesis is that it may extend the $N(x)$ events that end at w at time t. Another hypothesis is that it is just a newly established event. For example, as shown in the sub-figure(a) of Figure.3, the event (i_1, c_1) at time t_2 extends the event (u_1, i_1) at time t_2 while (u_1, i_2) is a new event occurring at time t_2. For the first hypothesis, based on the formula (2), we can learn that all the events intensity $\tilde{\lambda}(e)^T$ that end at w at time t has been calculated. Thus, we only need to calculate the event intensity of $N(x)$ newly established events $\tilde{\lambda}(x)^{T+1}$ at time t+1 and incrementally update the event intensity based on $\tilde{\lambda}(e)^T$ and $\tilde{\lambda}(x)^{T+1}$. For the second hypothesis, we calculate its intrinsic intensity in the formula (2) as the event intensity because it has no historical nodes. To this end, we can get all the intensity of new events at time t+1. Finally, we add them to the event intensity before time t+1 as the final event intensity. A toy example of incrementally updating is shown in sub-figure(d) of Figure.3. The incremental updating strategy is shown in Algorithm 1.

Algorithm 1 Incremental Updating Strategy

Input: x=$(w, v, r, t+1)$, $N(x)$, $\tilde{\lambda}(e)^{<=T}$
Output: Output the event intensity $\lambda(x)^{T+1}$
for event x **do**
  if x in $N(x)$ **then**
    Calculate its intrinsic intensity $\mu(x)$ via formula (3)
    Calculate its external intensity $\alpha_r(x)$ at time $t+1$ based on $\alpha_r(e)$ (formula (5)) and $\tilde{\lambda}_1$ (formula (7))
    Calculate its historical intensity $\kappa(t-t_p)$ via formula (12)
    Update event intensity $\tilde{\lambda}(x)^{T+1}$ via formula (2) based on $\tilde{\lambda}(e)^T$, $\mu(x)$, $\alpha_r(x)$ and $\kappa(t-t_p)$.
  else
    Calculate its intrinsic intensity $\mu(x)$ as event intensity $\tilde{\lambda}(x)^{T+1}$ via formula (3)
  end if
end for
for $T+1 \leftarrow 1$ **do**
  $\tilde{\lambda}(x)^{T+1} = \sum_{t=1}^{T+1} \tilde{\lambda}(x)^t$
end for
return $\lambda(x)^{T+1} = exp(\tilde{\lambda}(x)^{T+1})$

3.4. Aggregating Embedding for Recommendation

For the recommendation task, we generate the user embedding or item embedding at time t [18, 11] by multiplying and adding the intensity of historical events until time t with corresponding embedding one by one. Take user u as an example, its embedding o_u is defined as follows:

$$o_u = \sum_{p \in S_{u,<t}} \lambda(p) h_p,\ t = 1, 2, ..., T \quad (13)$$

where $S_{u,<t}$ is the nodes set before time t, h_p is the embedding of the event p. Similarly, the item embedding o_i can be generated based on the above process.

Finally, we put the user embedding o_u and item embedding o_i into an MLP to output the predicted clicking probability.

$$\hat{y}_{ui} = \sigma(o_u^T o_i) \tag{14}$$

where σ is the sigmoid function.

3.5. Model Optimization

Given the factor-driven event sequence $S_{u,<t}$ and $S_{i,<t}$ of target node u and source node i, the event intensity between u and i at time t can be computed as:

$$p(e_{u,i,r,t}|S_{u,<t}, S_{i,<t}) = \frac{\lambda(e_{u,i,r,t})}{\Sigma_{u' \in S_{u,<t}} \lambda(e_{u',i,r,t}) + \Sigma_{i' \in S_{i,<t}} \lambda(e_{u,i',r,t})} \tag{15}$$

where $\lambda(e_{u,i,r,t}) = exp(\tilde{\lambda}(e_{u,i,r,t}))$ denotes the positive intensity. Besides, we adopt negative sampling to accelerate training learning. Thus, the loss function [6, 26] can be written as follows:

$$\begin{aligned} &\mathcal{L}_{ep}(e) = \Sigma_{e \in E} log\sigma(\tilde{\lambda}(e)) - \\ &\Sigma_{m=1}^{M} \mathbb{E}_{i'} log\sigma(-\tilde{\lambda}(e_{i'})) - \Sigma_{m=1}^{M} \mathbb{E}_{u'} log\sigma(-\tilde{\lambda}(e_{u'})) \end{aligned} \tag{16}$$

where M is the size of negative samples, $\sigma(\cdot)$ is the sigmoid function.

Besides, for the recommendation task, we have the following unified loss function to capture the temporal embedding of the network:

$$\mathcal{L} = \mathcal{L}_{ep} + \omega \mathcal{L}_{rec} \tag{17}$$

where $\mathcal{L}_{rec}$ is the loss of recommendation. ω is the weight.

3.6. Complexity Analysis

For generating the factor-driven event sequence over time, the complexity is $O((U+I)N^t)$, for t=1, 2, ...T, where U and I are the numbers of users and items, N^t is the node number in each timestamp, and T is the maximum timestamps. In event intensity calculated by formula (2), the complexity of formula (3), formula (5), formula (7), and formula (12) are $O(d^2)$, $O(N^t R d^2)$, $O(N^t d^2)$, and $O(N^t)$, respectively, for t=1, 2, ...T, where d is the node embedding dimension, R is the number of the node type. Thus, the intensity measurement complexity of an event is $O(d^2+N^t R d^2)$. Therefore, the intensity measurement complexity of all events is $O((U+I)(d^2+N^t R d^2))$. If W new events occur, the complexity for incrementally updating these new events by Algorithm 1 is $O(WRd^2)$. Suppose the number of user-item pairs is Y, then the complexity of recommendation is $O(YTN^t d^2)$. To this end, the complexity of the proposed FMHP is $O((U+I)(d^2+N^t R d^2)+WRd^2+YTN^t d^2)$.

The time complexity is low in practice due to incremental updating. Since FMHP can cumulatively calculate event intensity over time, thus, we only need to calculate the intensities of the newly occurred events each time. Besides, the FMHP is easily paralleled. The events in each factor-driven event sequence can be processed in parallel since they are put into the multi-type attention. The number of N^t, T, and R are usually small constants, which can be ignored in practice.

4. Experiments

To evaluate the effectiveness of the proposed FMHP, we conduct extensive experiments to answer the following questions:

- RQ1. How does FMHP perform in the incremental recommendation compared with baselines?
- RQ2. How do three factors (intrinsic, external, and historical intensity) trigger the event generation?
- RQ3. What is the effect of external intensity of events on recommendation? Could FMHP provide a clue for the external intensity evolution of events?
- RQ4. Can multi-type temporal attention capture the external intensity for each event to enhance the recommendation?
- RQ5. Can the hybrid time decay function extract accurate historical intensity for each event to enhance the recommendation?
- RQ6. How do various hyper-parameters in FMHP affect the recommendation?

4.1. Experiments Settings

4.1.1. Datasets

To evaluate the effectiveness of FMHP, we conduct experiments on four widely used large datasets: Amazon-Beauty (Beauty)[1][46], Last-FM[2][3, 25], Movielens[3][16] and Amazon-Book (Book)[4][47]. The statistics of experimental datasets are in Table 2.

Beauty is a user-cosmetics interaction data with 5-core metadata. To construct a factor-driven event sequence, we first extract the brand, category, and feature node from the meta-data. We then generate the events (e.g., I-B, I-C, I-F), where the time of these events is the next timestamp of its linked nodes. Finally, we treat the rating of 5 as positive and select users with more than 50 interactions to learn enough for modeling users.

LastFM is user-music interaction data. We construct the factor-driven event sequence using U-M, U-F, F-M, and G-M in the data. We select users with more than 40 interactions for training. As U-F and G-M have no timestamps, we set the time of these events as the next timestamp of its linked nodes.

Movielens is a user-movie interaction data with four event types (e.g., U-M, M-D, M-A, and M-G). If the user's rating on a movie is more significant than 10, it is an implicit event. We delete the user with less than 30 interactions and construct factor-driven event sequences using the above events.

The book contains user-movie interaction data and a 5-core meta-data. We estimate users interacting with books less than 40 interactions and regard the rating 5 as positive. Like Beauty, we choose B-W and B-C from meta-data and construct the factor-driven event sequence.

[1] https://jmcauley.ucsd.edu/data/amazon/
[2] https://grouplens.org/datasets/hetrec-2011/
[3] https://grouplens.org/datasets/movielens/20m/
[4] https://jmcauley.ucsd.edu/data/amazon/

Table 2: Experimental statistics of the four datasets

Datasets	Node Type	Nodes	Event Type	Events	Timestamps
Beauty	User(U) Item(I) Brand(B) Category(C) Feature(F)	6710 2753 334 22 7362	U-I I-B I-C I-F	85598 148576 148576 196740	6
LastFM	User(U) Genre(G) Music(S) Friends(F)	1892 30 16524 1892	U-S U-F F-S G-S	92834 25434 25434 186479	6
Movielens	User(U) Director(D) Actor(A) Genre(G) Movie(M)	2113 4053 95241 20 9407	U-M M-D M-A M-G	231742 65132 65132 20809	6
Book	User(U) Writer(W) Book(B) Category(C)	5264 18181 9160 22	U-B B-W B-C	67960 145610 9160	6

4.1.2. Baselines and Evaluation Metrics

To verify the effectiveness of the FMHP, We employ two static recommendation models (LINE and DeepWalk), one dynamic recommendation model (LightGCN), and four Hawkes-based incremental models (HTNE, M^{2}DNE, HPGE, MHNE). The detailed description is listed as follows. We use Hit Ratio (HR@k), Normalized Discounted Cumulative Gain (NDCG@k), and Recall (Recall@k) under top k circumstance as our evaluation metrics for the top-k recommendation. Specifically, we set k = {5, 10, 15, 20} through our experiments.

- **LINE** [33]: LINE learns latent representation for the large-scale network by preserving first- and higher-order proximity. We use higher-order proximity as our baseline.
- **DeepWalk** [30]: DeepWalk utilizes truncated random walks to learn latent representations.
- **LightGCN** [14]: LightGCN designs a simplified Graph Convolutional Network (GCN) to dynamically aggregate neighborhood embedding for the recommendation.
- **HTNE** [48]: HTNE integrates the Hawkes process with an attention mechanism to capture the effect of historical events on the current event.
- **M^{2}DNE** [25]: M^{2}DNE models graph evolution at micro- and macro- level using Hawkes process. We only use its micro-level as our baseline.
- **HPGE** [16]: HPGE utilizes the Hawkes process to capture the excitation of historical events regarding heterogeneity and dynamics.
- **MHNE** [3]: MHNE encodes multi-aspect embedding of historical nodes into Hawkes process to assign the weights of each aspect.

4.1.3. Hyperparameter Settings

For Line, we use the high-order proximity to learn node embedding for a fair comparison since the length of the factor-driven event sequence used in our method is bigger than 1. In terms of DeepWalk, the walk length for each node is set to 100. For LightGCN, the neighborhood aggregation number for each node is 100. The history length or meta-path length in HTNE, M^{2}DNE, HPGE, and MHNE is the same as the length of factor-driven event sequence in our method, where we set the maximum timestamp $T = 3$ for Beauty and Book and $T = 4$ for LastFM and Movielens. The other parameters of baselines are set by default.

For our method, we set the embedding dimension $d = 128$ and the batch size as 1024. The learning rate is selected from $\{0.001, 0.0005, 0.0001\}$, the regularization weight $\omega = \{0.3, 0.4, 0.4, 0.3\}$ for Beauty, LastFM, Movielens and Book, respectively. The negative sampling size is 10 (if any). We perform standard 10-fold cross-validation for all models on all datasets. We adopt a mini-batch Adam optimizer. We set the training, evaluation, and test ratio for all four datasets as 6:2:2. We use Pytorch to implement our method. Our software environments are Pytorch (1.11.0), torch-geometric (2.2.0), torch-cluster (1.6.0), torch-scatter (2.0.9), and torch-sparse (0.6.14). Moreover, all models are trained on Nvidia RTX A5000 with Intel(R) Xeon(R) E-2278G CPU @ 3.40GHz.

4.2. Overall Performance (RQ1)

In this section, we evaluate the performance of our proposed FMHP model on four datasets and compare it with seven baselines for the top-k recommendation. The results are in Table 3. Besides, we reported the improvements and statistical significance test in Table 3, which are calculated between our proposed FMHP method and the best incremental recommendation baseline MHNE (highlighted with underline). We set a p-value <0.05 as a significant improvement.

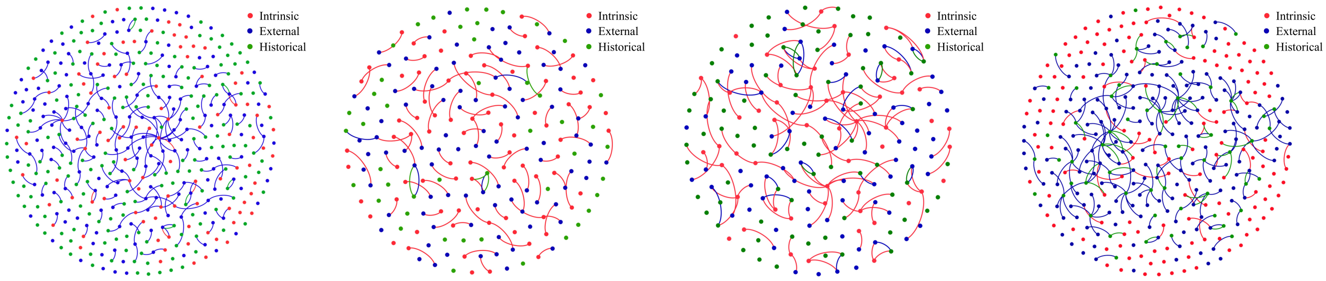
As shown in Table 3, our proposed FMHP performs significantly better than the baselines from all evaluation metrics. Specifically, the average improvements of FMHP over the strongest incremental recommendation baseline MHNE *w.r.t* HR@k are 9.77%, 9.35%, 9.32%, 10.10%, *w.r.t* NDCG@k are 8.86%, 10.26%, 9.81%, 9.38%, and *w.r.t* Recall@k are 9.64%, 9.32%, 8.97%,9.85% in Beauty, LastFM, MovieLens, and Book, respectively. We credit this better performance for two reasons. First, except for the simple time interval used in MHNE, FMHP further extracts the accumulated occurrence frequency of historical events as a pseudo time of each historical node, which can capture the evolution distribution of the network over time in a more fine-grained manner. Second, FMHP considers the historical events' type-wise semantics and temporal effect to calculate the event intensity. In contrast, MHNE only considers the node types, limiting the representation of incremental recommendations. Concerning p-value, as shown in Table 3, all p-values of our proposed FMHP compared with the best incremental recommendation baseline MHNE (highlighted with underline) are less than 0.05. This case demonstrates the improvements of FMHP over the best incremental recommendation baseline are statistically significant.

Compared with other incremental recommendation methods (e.g., HPGE, M^{2}DNE, HTNE), the superior performance of FMHP is due to its excavation for underlying factors of each

Table 3: Overall performance on incremental recommendation compared with baselines. Improve and p-value denote the relative improvements(%) and t-test results of FMHP compared with MHNE, respectively.

Datasets	Metrics	LINE	DeepWalk	LightGCN	HTNE	M^{2}DNE	HPGE	MHNE	**FMHP**	*Improve*	*p-value*
Beauty	HR@5	0.1575	0.2104	0.2314	0.4513	0.5678	0.5781	0.5934	**0.6872**	*9.38%*	*5.90e-6*
	HR@10	0.2003	0.2472	0.2562	0.5542	0.6557	0.6830	0.7029	**0.8016**	*9.87%*	*5.95e-6*
	HR@15	0.2179	0.2821	0.2852	0.5817	0.6923	0.6990	0.7210	**0.8152**	*9.42%*	*6.36e-6*
	HR@20	0.2372	0.2953	0.3010	0.6184	0.7057	0.7123	0.7311	**0.8353**	*10.42%*	*2.68e-8*
	NDCG@5	0.1321	0.2000	0.1849	0.4175	0.5293	0.5318	0.5473	**0.6301**	*8.28%*	*2.20e-5*
	NDCG@10	0.1799	0.2072	0.2159	0.5337	0.6210	0.6429	0.6552	**0.7418**	*8.66%*	*1.87e-5*
	NDCG@15	0.1872	0.2184	0.2301	0.5446	0.6464	0.6692	0.6717	**0.7630**	*9.13%*	*5.35e-6*
	NDCG@20	0.2011	0.2321	0.2446	0.5621	0.6501	0.6818	0.6982	**0.7917**	*9.35%*	*7.64e-5*
	Recall@5	0.1723	0.2345	0.2454	0.4721	0.5822	0.5851	0.6124	**0.6957**	*8.33%*	*4.60e-5*
	Recall@10	0.2287	0.2513	0.2635	0.5632	0.6631	0.7065	0.7131	**0.8104**	*9.73%*	*1.06e-7*
	Recall@15	0.2315	0.2622	0.2958	0.5943	0.7101	0.7136	0.7359	**0.8399**	*10.40%*	*1.80e-6*
	Recall@20	0.2462	0.2734	0.3210	0.6302	0.7152	0.7224	0.7428	**0.8436**	*10.08%*	*5.43e-5*
LastFM	HR@5	0.1093	0.1442	0.1584	0.4023	0.5021	0.5149	0.5362	**0.6304**	*9.42%*	*4.95e-7*
	HR@10	0.1402	0.1856	0.2001	0.5050	0.6029	0.6273	0.6374	**0.7220**	*8.64%*	*5.56e-5*
	HR@15	0.1547	0.2095	0.2148	0.5131	0.6154	0.6320	0.6621	**0.7577**	*9.56%*	*7.86e-6*
	HR@20	0.1638	0.2273	0.2321	0.5298	0.6237	0.6392	0.6773	**0.7752**	*9.79%*	*4.68e-6*
	NDCG@5	0.0856	0.1407	0.1441	0.3298	0.4465	0.4678	0.4721	**0.5363**	*9.42%*	*5.74e-5*
	NDCG@10	0.1032	0.1747	0.1859	0.4329	0.5217	0.5449	0.5683	**0.6674**	*9.91%*	*1.15e-6*
	NDCG@15	0.1346	0.2019	0.2035	0.4478	0.5400	0.5771	0.5824	**0.6917**	*10.93%*	*1.85e-8*
	NDCG@20	0.1575	0.2150	0.2216	0.4690	0.5412	0.5800	0.5953	**0.7029**	*10.76%*	*2.08e-5*
	Recall@5	0.1187	0.1621	0.1731	0.4123	0.5267	0.5329	0.5458	**0.6371**	*9.13%*	*1.35e-7*
	Recall@10	0.1621	0.1937	0.2154	0.5174	0.6114	0.6317	0.6521	**0.7442**	*9.21%*	*2.78e-6*
	Recall@15	0.1754	0.2221	0.2469	0.5319	0.6362	0.6471	0.6704	**0.7687**	*9.83%*	*1.24e-5*
	Recall@20	0.1851	0.2419	0.2521	0.5478	0.6453	0.6674	0.6851	**0.7763**	*9.12%*	*7.45e-7*
Movielens	HR@5	0.1126	0.2041	0.2137	0.4276	0.5223	0.5451	0.5732	**0.6650**	*9.18%*	*2.21e-5*
	HR@10	0.1596	0.2407	0.2655	0.5121	0.6176	0.6380	0.6657	**0.7609**	*9.52%*	*5.52e-5*
	HR@15	0.1824	0.2598	0.2747	0.5314	0.6254	0.6593	0.6981	**0.7881**	*9.00%*	*9.80e-6*
	HR@20	0.1975	0.2625	0.2845	0.5472	0.6389	0.6852	0.7033	**0.7989**	*9.56%*	*5.73e-6*
	NDCG@5	0.1071	0.1441	0.1498	0.4089	0.4623	0.4877	0.5201	**0.6113**	*9.12%*	*3.11e-7*
	NDCG@10	0.1245	0.1997	0.2072	0.4929	0.5769	0.5872	0.6098	**0.7065**	*9.67%*	*3.74e-6*
	NDCG@15	0.1380	0.2103	0.2183	0.5174	0.5893	0.6128	0.6300	**0.7269**	*9.69%*	*2.60e-5*
	NDCG@20	0.1482	0.1482	0.2369	0.5248	0.6123	0.6274	0.6399	**0.7474**	*10.75%*	*2.25e-6*
	Recall@5	0.1320	0.2135	0.2240	0.4352	0.5326	0.5521	0.5817	**0.6721**	*9.04%*	*7.53e-9*
	Recall@10	0.1719	0.2583	0.2705	0.5217	0.6219	0.6471	0.6834	**0.7732**	*8.89%*	*9.70e-5*
	Recall@15	0.1934	0.2721	0.2841	0.5449	0.6417	0.6723	0.6974	**0.7958**	*8.85%*	*5.36e-5*
	Recall@20	0.2129	0.2781	0.2921	0.5602	0.6503	0.6911	0.7123	**0.8034**	*9.11%*	*5.01e-5*
Book	HR@5	0.1954	0.2517	0.2706	0.5227	0.6024	0.6158	0.6228	**0.7235**	*10.07%*	*6.73e-11*
	HR@10	0.2171	0.2942	0.3095	0.6185	0.7002	0.7110	0.7167	**0.8102**	*9.35%*	*8.99e-6*
	HR@15	0.2211	0.3158	0.3219	0.6374	0.7119	0.7331	0.7422	**0.8413**	*10.91%*	*9.73e-8*
	HR@20	0.2382	0.3321	0.3416	0.6449	0.7234	0.7442	0.7510	**0.8517**	*10.07%*	*4.52e-6*
	NDCG@5	0.1420	0.1859	0.2021	0.4815	0.5458	0.5676	0.5883	**0.6784**	*8.98%*	*8.18e-8*
	NDCG@10	0.1645	0.2534	0.2655	0.5492	0.6317	0.6425	0.6530	**0.7522**	*9.92%*	*9.87e-5*
	NDCG@15	0.1769	0.2624	0.2774	0.5723	0.6550	0.6719	0.6804	**0.7805**	*9.92%*	*7.24e-6*
	NDCG@20	0.1827	0.2781	0.2900	0.5947	0.6729	0.6729	0.7031	**0.7900**	*8.69%*	*2.37e-7*
	Recall@5	0.2100	0.2631	0.2814	0.5315	0.6117	0.6241	0.6317	**0.7321**	*10.04%*	*7.10e-8*
	Recall@10	0.2251	0.3119	0.3221	0.6277	0.7103	0.7223	0.7324	**0.8301**	*9.77%*	*2.90e-7*
	Recall@15	0.2320	0.3276	0.3302	0.6482	0.7221	0.7417	0.7530	**0.8517**	*9.87%*	*4.60e-6*
	Recall@20	0.2518	0.3452	0.3527	0.6536	0.7324	0.7599	0.7731	**0.8702**	*9.71%*	*4.08e-6*

event generation (the intensity, external, and historical intensity). In contrast, other incremental recommendation methods only use one of these factors as additional features for an event. Besides, all incremental recommendation methods perform better than the dynamic recommendation model (e.g., LightGCN). This phenomenon verifies that tracking the formation of events sequence (incremental recommendation methods) can capture more accurate event embedding than treating each event as a static graph (dynamic recommendation model). Moreover, the static models (e.g., LINE and DeepWalk) underperform on our dynamic graph datasets. LINE and DeepWalk perform poorly on all datasets, indicating that random sampling of static methods is insufficient to capture the similarity evolution in the dynamic graph.



(a) Beauty (b) LastFM (c) MovieLens (d) Book

Figure 4: The events generations triggered by three factors on four datasets

4.3. Case Study of Events Generation Triggered by Three Factors (RQ2)

In order to determine which factor has the dominant influence on event generation, we conducted a case study of event generation in four datasets. We randomly sample 0.1% to show the results, as drawing millions of events in each dataset is impractical. As depicted in Figure.4, each point represents an event, and its color represents the factor type that triggers this event. If an event triggers another due to a factor, we add a line of the same color as the factor to another.

As shown in Figure.4, the distribution of dominant factors varies among the different datasets. In LastFM and Movielens, intrinsic intensity dominates the user's events. This phenomenon indicates that people prefer music and movies that align with their interests. After all, in real life, peoples' preferences for music or movie generally do not change. Besides, in Beauty and Book datasets, the external intensity dominates event generation. This case is because our selection for cosmetics or books depends on the requirements at that time. Thus, external factors such as cosmetic brands or book categories may always affect users. The historical intensity also plays a role in Beauty and Book, as users' prior experiences can help their selections. Overall, FMHP can determine the dominant reasons behind users' every event.

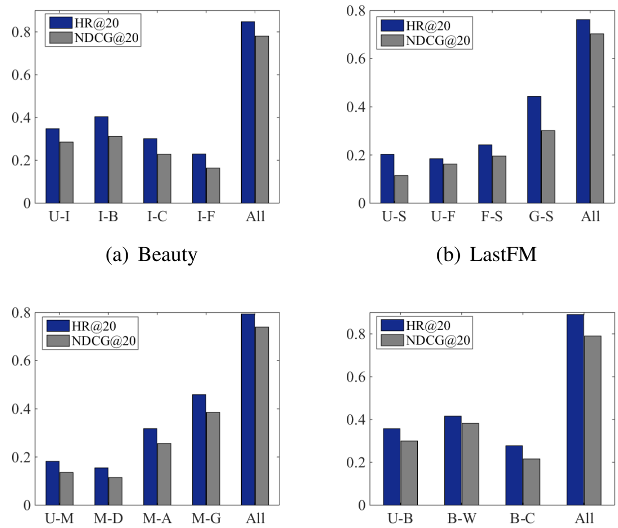
4.4. Analysis on External Intensity of Events (RQ3)

To investigate the role of the external intensity of the event, we design the following two experiments. (1) the effect of each event's external intensity on the recommendation. (2) visualization of external intensity evolution of events.

4.4.1. Analysis on Effect of External Intensity for Recommendation

To evaluate the external intensity effect of the event, we only adopt one type of event at each time to calculate the event intensity and use the embedding of the final timestamp for the recommendation. We use HR@20 and NDCG@20 as the evaluation metrics. The experiment averages five times the experiments as outputs. The results are shown in Figure.5.

As shown from Figure.5, the embedding with all types of events achieves the best performance *w.r.t* HR@20 and NDCG@20. As shown in all sub-figures, the recommendation



(a) Beauty (b) LastFM (c) Movielens (d) Book

Figure 5: The effect of external intensity of events on recommendation

with only one type event decreases significantly *w.r.t* HR@20 and NDCG@20. It indicates that the user's events are determined by multiple factors (such as color, properties, brand et al.) rather than a single one, which verified the effectiveness of type-wise attention in our method. Additionally, the importance of each event type, as indicated by the height of the bars, reveals the user's shopping habits. For example, in the LastFM dataset, G-S has the highest bar, meaning this user prefers items of the same genre.

4.4.2. Visualization of External Intensity Evolution on Events

In this section, we visualize the evolution of the external intensity of events over time, which is the product of type-wise attention in formula (5) and temporal attention in formula (7). We take user 1682 from LastFM and 44285 from Movielens as the experimental subject, respectively.

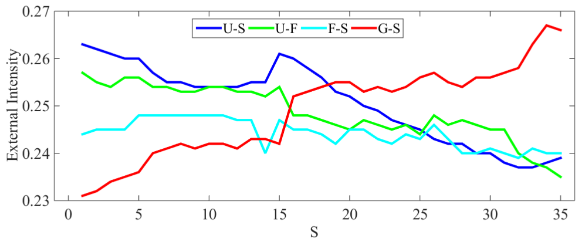
As shown in Figure.6, the external intensity of each event type changes over time. For user 1682 from LastFM, as shown in the first subgraph, the purse line (U-S), the green line (U-F), and the blue line (F-S) have a clear downward trend, while the red line (G-S) grows over time. For user 44285 from Movie-

Table 4: Ablation study of multi-type temporal attention of external intensity on incremental recommendation

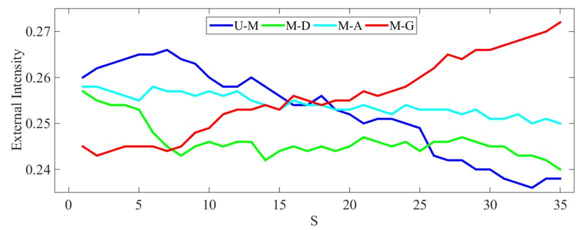
Datasets	Beauty		LastFM		Movielens		Book	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
No-attention	0.4017	0.3520	0.3824	0.3351	0.3341	0.3049	0.4552	0.4223
Type-wise attention	0.7837	0.7232	0.7281	0.6997	0.7354	0.6793	0.8246	0.7265
Temporal attention	0.6958	0.6372	0.6208	0.5915	0.6479	0.6138	0.7125	0.6845
All attention	**0.8353**	**0.7817**	**0.7752**	**0.7029**	**0.7989**	**0.7474**	**0.8517**	**0.7900**

Table 5: Ablation study of hybrid time decay function of historical intensity on incremental recommendation

Datasets	Beauty		LastFM		Movielens		Book	
	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
No-time	0.2019	0.1733	0.1445	0.1036	0.1582	0.1291	0.1892	0.1547
Pseudo-time	0.6233	0.5521	0.5472	0.5018	0.5784	0.5247	0.6501	0.5739
True-time	0.7628	0.7045	0.7094	0.6862	0.7213	0.6581	0.8021	0.7145
All-time	**0.8353**	**0.7817**	**0.7752**	**0.7029**	**0.7989**	**0.7474**	**0.8517**	**0.7900**



(a) LastFM



(b) Movielens

Figure 6: Visualization of external intensity evolution of events

lens, as shown in the second subgraph, the blue line (U-M) decreases, while the red line (M-G) has a rising trend. These two cases demonstrate that users may not initially have specific music or movie preferences and choose under various investigations (such as querying singers and asking friends et al.). When he develops a taste for music over time, his selection gradually synchronizes with his intrinsic interests. These cases demonstrate that our method can reveal the evolving effect of different types of events and provides valuable information for incremental recommendation.

4.5. *Ablation Study on Multi-type Temporal Attention of External Intensity (RQ4)*

To assess the impact of multi-type temporal attention, we compared it with three model variants: no-attention, type-wise attention, and temporal attention. To evaluate the effectiveness of the type-wise attention, we replaced the formula (5) with a concatenation representation of two nodes. We removed temporal attention from the formula (7) to evaluate its effectiveness. We use HR@20 and NDCG@20 as our evaluation metrics. The results are shown in Table 4.

As seen in Table 4, the model with all attention performs the best in the incremental recommendation. Both the type-wise attention variant and temporal attention variant outperform the no-attention variant. This phenomenon suggests that the type-wise semantics extracted by the type-wise attention and the time decay effect obtained by the temporal attention are beneficial for representing the dynamic network and improving the incremental recommendation. In addition, the type-wise attention variant outperforms the temporal attention variant by an average of 9.87% and 7.54% *w.r.t* HR@20 and NDCG@20, respectively. It may be because the type-wise semantics of nodes provide a more fine-grained representation of the user's interests. In contrast, the time effect can only roughly represent the time interval.

4.6. *Ablation Study on Hybrid Time Decay Function of Historical Intensity (RQ5)*

To evaluate the impact of the hybrid time decay function in historical intensity, we compare it with three model variants: no-time, pseudo-time, and true-time. We use HR@20 and NDCG@20 as the evaluation metrics and present the results in Table 5.

The table shows that the all-time variant performs the best, while the no-time variant performs the worst on all datasets. The pseudo-time and true-time variants outperform the no-time variant, demonstrating their effectiveness. The pseudo-time variant only has a 21% reduction in performance compared to the all-time variant, indicating that it can capture the user's temporal interests through the frequency of each node. Additionally, the pseudo-time and true-time variants have different contributions to the incremental recommendation, with the true-time variant showing a 13% higher recommendation performance than the pseudo-time variant. This phenomenon may be because the true-time variant represents an exact time relevant to the user's interaction, which more closely synchronizes with the user's historical interaction time.

4.7. Analysis on Hyper-parameters Sensitivity (RQ6)

To verify the effect of different sampling times T in the factor-driven event sequence, we conduct incremental recommendations under the $T = \{1, 2, 3, 4\}$. The results are shown in Table 6 w.r.t HR@20 and NDCG@20.

As we can see from Table 6, our method achieves the best recommendation performance when $t = 2$, t=2, $t = 3$, and $t = 2$ *w.r.t* HR@20 and $t = 2$, $t = 3$, $t = 3$, and $t = 2$ *w.r.t* NDCG@20 in Beauty, LastFM, Movielens, and Book, respectively. FMHP has the worst recommendation performance with $t = 1$ in all datasets *w.r.t* HR@20 and *w.r.t* NDCG@20. A reasonable suspicion is that if the sampling time is too short, it can barely explore the historical attributes of the current event. When $t = 4$, its performance averagely decreases 1.58% *w.r.t* HR@ 20 and 1.69% *w.r.t* HDCG than the results when $t = 3$. It is because a big sampling time may bring noise or interferential information, which encourages us to choose suitable timestamps for each dataset.

Table 6: Recommendation results of FMHP with different sample timestamps in factor-driven event sequence

Metrics	Datasets	Beauty	LastFM	Movielens	Book
HR@20	$t = 1$	0.8149	0.7548	0.7839	0.8303
	$t = 2$	0.8353	0.7621	0.7936	0.8517
	$t = 3$	0.8310	0.7750	0.7972	0.8472
	$t = 4$	0.8242	0.7620	0.7554	0.8455
NDCG@20	$t = 1$	0.7526	0.6538	0.7041	0.7580
	$t = 2$	0.7805	0.6922	0.7392	0.7900
	$t = 3$	0.7723	0.7026	0.7472	0.7784
	$t = 4$	0.7698	0.6731	0.7269	0.7631

5. Related Work

We review the incremental recommendations from three factors (intrinsic, external, and historical) in our method. Table 7 summarises the literature regarding these three factors.

Based on the user's historical behaviors sequences (intrinsic factor), some incremental recommendation works [37, 40, 8, 28] split a dynamic graph into several snapshots and integrate deep auto-encoders and sequential neural networks to learn the evolving embeddings. For example, EvolveGCN [28] captures the dynamics of the user's behavior through an RNN to evolve the GCN extract features without resorting to node time. RSL-GRU [8] extract snapshot features by rewarding multiple paths using reinforcement learning and use Gated Recurrent Unit (GRU) to capture the temporal evolution of the graph. However, these methods have some limitations. (1) The semantics of the user's behaviors are extracted and evolved with the static snapshot, limiting the expression of nodes for incremental recommendations. (2) The time of each node in these methods is consistent with the snapshot it belongs to, which deviates from its real formation time considered in our method.

To enrich the explicit semantics of nodes and edges in incremental recommendations, some work [13, 15, 23, 1] leverages semantic information (external factor) in the network. Specifically, WSHE [15] utilizes meta-path-based similarity based on user feedback to measure the user's preference for a recommendation task. BAR [13] adopt users' sequential heterogeneous one-class feedback (item, behavior) to models (e.g., Recurrent Neural Network (RNN), Convolutional Neural Network (CNN) or Graph Neural Network (GNN)) for the recommendation. GHCF [4] takes advantage of GCN and further improves it to jointly embed representations of users, items, and relations with non-sampling optimization. MAGNN [9] proposes a GNN-based model to extract intra- and inter-meta-path connectivity among different types of nodes to generate user and item embedding for the recommendation. However, these methods integrate all meta-paths from static snapshots as the semantics of the user or item, blurring semantic deviations for the user or item. Besides, the temporal information of each node in meta-paths is also not considered.

To overcome the above limitation, some methods [48, 16, 25] model the evolution of temporal networks by tracking the neighborhood formation of each node. AMHP [35] utilizes the Hawkes to model the recurrence of the same items in the user sequence as a self-excitation process and capture its short-term and long-term time evolution using an RNN. HPGE [16] designs a heterogeneous evolved attention mechanism with the Hawkes process to extract the edge formation. MHNE [3] proposes a mixed Hawkes process to explore multi-aspect nodes that drive the neighborhood formation. However, these methods model the generation of each node with limited user sequences or meta-paths. None of them have analyzed the underlying factors that trigger the event generation.

Table 7: The evaluation and summarization of all literature reviews in terms of three factors considers in our FMHP method

Methods	Intrinsic	External	Historical
MFGAT [40]	×	✓	×
KGCR [37]	✓	✓	×
EvolveGCN [28]	×	✓	×
RSL-GRU [8]	×	✓	×
BAR [13]	✓	×	×
WSHE [15]	✓	✓	×
HS-GCN [23]	×	✓	×
GHCF [4]	✓	✓	×
MAGNN [9]	×	✓	×
GraphSAIL [1]	×	✓	×
AMHP [4]	✓	×	✓
HTNE [48]	✓	×	✓
M^2DNE [25]	✓	×	✓
HPGE [16]	×	✓	✓
MHNE [3]	×	✓	✓

6. Conclusion and Future Work

We propose an FMHP for the event-based incremental recommendation, which models the underlying factors that trigger the generation of each specific event and incrementally updates the event formation as new events occur. Specifically, we consider three factors that may trigger event generation: intrinsic, external, and historical intensity. An intrinsic intensity function, a multi-type temporal function, and a hybrid time decay function are incorporated into FMHP to calculate the intrinsic, external, and historical intensity, respectively. In addition, FMHP uses an incremental updating strategy to calculate the event intensity incrementally. We conduct extensive experiments on four public datasets (e.g., Amazon Beauty, LastFM, Movielens, and Amazon Book). Compared with state-of-the-art incremental recommendation methods, our proposed FMHP model achieves superior performance, averagely, with up to 9.64% improvement in hit rate, 9.57% in normalized discounted cumulative gain, and 9.45% in the recall. Besides, the case study shows that the three factors in our proposed FMHP method play a vital role in triggering event generation.

For future work, we will consider improving our method to deal with more complicated temporal patterns, such as metagraphs, a potential research field. Although the Hawkes process achieves outstanding performance in capturing the stimulating effects of the event, we will explore some deep neural networks (e.g., GCN and GNN), which can capture the structural and semantics information of heterogeneous graphs. We will also expand our method to model the occurrence of an event from the fairness of multiple stakeholders, such as the promotions of the merchant, the friendship of users, and the review of the item.

Acknowledgements

The authors would like to acknowledge the support provided by the Innovation Method Project of Ministry of Science and Technology, China under Grant 2020IM020100, the National Natural Science Foundation of China under Grant 61872222, the Key Research and Development Program of Shandong Province (2020CXGC010102), the project ZR2020LZH011 supported by Shandong Provincial Natural Science Foundation and the financial support from China Scholarship Council.

References

[1] Ahrabian, K., Xu, Y., Zhang, Y., Wu, J., Wang, Y., Coates, M., 2021. Structure aware experience replay for incremental learning in graph-based recommender systems, in: CIKM, ACM, Queensland, Australia. pp. 2832–2836.

[2] An, J., Li, G., Jiang, W., 2022. Thcore: Things of interest recommendation based on novel things correlations. Inf. Sci. 605, 317–332.

[3] Chang, Y., Liu, G., Zuo, Y., Wu, J., 2021. Multi-aspect temporal network embedding: A mixture of hawkes process view. arXiv preprint arXiv:2105.08566 abs/2105.08566.

[4] Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y., Ma, S., 2021a. Graph heterogeneous multi-relational recommendation, in: AAAI, AAAI Press, Virtual Event. pp. 3958–3966.

[5] Chen, Z., Zhang, W., Yan, J., Wang, G., Wang, J., 2021b. Learning dual dynamic representations on time-sliced user-item interaction graphs for sequential recommendation, in: CIKM, ACM, Virtual Event, Queensland, Australia. pp. 231–240.

[6] Chiang, Y., Liu, Y., Tsai, C., Lou, J., Tsai, M., Wang, C., 2022. Recdelta: An interactive dashboard on top-k recommendation for cross-model evaluation, in: SIGIR, ACM, Madrid, Spain. pp. 3224–3228.

[7] Cho, J., Hyun, D., Kang, S., Yu, H., 2021. Learning heterogeneous temporal patterns of user preference for timely recommendation, in: WWW, ACM / IW3C2, Ljubljana, Slovenia. pp. 1274–1283.

[8] Cui, Z., Chen, H., Cui, L., Liu, S., Liu, X., Xu, G., Yin, H., 2022. Reinforced kgs reasoning for explainable sequential recommendation. World Wide Web 25, 631–654.

[9] Fu, X., Zhang, J., Meng, Z., King, I., 2020. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding, in: WWW, ACM / IW3C2, Taipei. pp. 2331–2341.

[10] Ge, H., Yang, C., Xu, Y., 2022. Incremental updating three-way regions with variations of objects and attributes in incomplete neighborhood systems. Inf. Sci. 584, 479–502.

[11] Guo, P., Ye, Z., Xiao, K., Zhu, W., 2022. Weighted aggregating stochastic gradient descent for parallel deep learning. IEEE Trans. Knowl. Data Eng. 34, 5037–5050.

[12] Guo, S., Wang, Y., Yuan, H., Huang, Z., Chen, J., Wang, X., 2021. TAERT: triple-attentional explainable recommendation with temporal convolutional network. Inf. Sci. 567, 185–200.

[13] He, M., Pan, W., Ming, Z., 2022. BAR: behavior-aware recommendation for sequential heterogeneous one-class collaborative filtering. Inf. Sci. 608, 881–899.

[14] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M., 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation, in: SIGIR, ACM, China. pp. 639–648.

[15] Hu, B., Wang, H., Wang, L., 2021. WSHE: user feedback-based weighted signed heterogeneous information network embedding. Inf. Sci. 579, 167–185.

[16] Ji, Y., Jia, T., Fang, Y., Shi, C., 2021. Dynamic heterogeneous graph embedding via heterogeneous hawkes process, in: Machine Learning and Knowledge Discovery in Databases, Springer, Bilbao, Spain. pp. 388–403.

[17] Khan, T.M., Robles-Kelly, A., 2020. A derivative-free method for quantum perceptron training in multi-layered neural networks, in: ICONIP, Springer, Bangkok, Thailand. pp. 241–250.

[18] Leng, Y., Yu, L., Niu, X., 2022. Dynamically aggregating individuals' social influence and interest evolution for group recommendations. Inf. Sci. 614, 223–239.

[19] Li, H., Wang, Y., Lyu, Z., Shi, J., 2022. Multi-task learning for recommendation over heterogeneous information network. IEEE Trans. Knowl. Data Eng. 34, 789–802.

[20] Liang, T., Ma, L., Zhang, W., Xu, H., Xia, C., Yin, Y., 2022. Content-aware recommendation via dynamic heterogeneous graph convolutional network. Knowl. Based Syst. 251, 109185.

[21] Liang, Y., Huang, C., Bao, X., Xu, K., 2021. Sequential dynamic event recommendation in event-based social networks: An upper confidence bound approach. Inf. Sci. 542, 1–23.

[22] Liu, H., Guo, L., Li, P., Zhao, P., Wu, X., 2021. Collaborative filtering with a deep adversarial and attention network for cross-domain recommendation. Inf. Sci. 565, 370–389.

[23] Liu, H., Wei, Y., Yin, J., Nie, L., 2023. HS-GCN: hamming spatial graph convolutional networks for recommendation. CoRR abs/2301.05430.

[24] Liu, S., Chen, Z., Li, X., 2019. Time-semantic-aware poisson tensor factorization approach for scalable hotel recommendation. Inf. Sci. 504, 422–434.

[25] Lu, Y., Wang, X., Shi, C., Yu, P.S., Ye, Y., 2019. Temporal network embedding with micro-and macro-dynamics, in: CIKM, ACM, Beijing, China. pp. 469–478.

[26] Luo, F., Wu, J., Wang, T., 2022. Discrete listwise personalized ranking for fast top-n recommendation with implicit feedback, in: IJCAI, ijcai.org, Vienna, Austria. pp. 2159–2165.

[27] Niu, X., Li, B., Li, C., Tan, J., Xiao, R., Deng, H., 2021. Heterogeneous graph augmented multi-scenario sharing recommendation with tree-guided expert networks, in: WSDM, ACM, Israel. pp. 1038–1046.

[28] Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T., Leiserson, C., 2020. Evolvegcn: Evolving graph convolutional networks for dynamic graphs, in: Proceedings of the AAAI conference on artificial intelligence, pp. 5363–5370.

[29] Patro, G.K., Chakraborty, A., Ganguly, N., Gummadi, K.P., 2020. Fair updates in two-sided market platforms: On incrementally updating recommendations, in: IAAI, AAAI Press, New York, NY, USA. pp. 181–188.

[30] Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: Online learning of social representations, in: SIGKDD, ACM, New York, NY, USA. pp. 701–710.

[31] Qi, L., Zhang, X., Li, S., Wan, S., Wen, Y., Gong, W., 2020. Spatial-temporal data-driven service recommendation with privacy-preservation. Inf. Sci. 515, 91–102.

[32] Shen, Y., Feng, Y., Fang, B., Zhou, M., Kwong, S., Qiang, B., 2020. DSRPH: deep semantic-aware ranking preserving hashing for efficient multi-label image retrieval. Inf. Sci. 539, 145–156.

[33] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q., 2015. LINE: large-scale information network embedding, in: WWW, ACM, Florence, Italy. pp. 1067–1077.

[34] Wang, W., Ma, H., Zhao, Y., Yang, F., Chang, L., 2022. SEEP: semantic-enhanced question embeddings pre-training for improving knowledge tracing. Inf. Sci. 614, 153–169.

[35] Wen, X., Wang, J., Yang, X., Zhang, Q., 2021. A sequential recommendation of hawkes process fused with attention mechanism, in: HPCC/DSS/SmartCity/DependSys, IEEE. pp. 301–308.

[36] Xia, J., Li, D., Gu, H., Liu, J., Lu, T., Gu, N., 2022. FIRE: fast incremental recommendation with graph signal processing, in: WWW, ACM, Virtual Event, Lyon, France. pp. 2360–2369.

[37] Xiao, C., Sun, L., Ji, W., 2020. Temporal knowledge graph incremental construction model for recommendation, in: Web and Big Data, Springer, Tianjin, China. pp. 352–359.

[38] Xie, T., Xu, Y., Chen, L., Liu, Y., Zheng, Z., 2021. Sequential recommendation on dynamic heterogeneous information network, in: ICDE, IEEE, Chania, Greece. pp. 2105–2110.

[39] Yuan, W., Wang, H., Yu, X., Liu, N., Li, Z., 2020. Attention-based context-aware sequential recommendation model. Inf. Sci. 510, 122–134.

[40] Zhang, J., Liang, S., Deng, Z., Shao, J., 2021. Spatial-temporal attention network for temporal knowledge graph completion, in: DASFAA, Springer, Taibei,China. pp. 207–223.

[41] Zhang, R., Walder, C.J., Rizoiu, M., 2020. Variational inference for sparse gaussian process modulated hawkes process, in: AAAI, AAAI Press, New York, NY, USA. pp. 6803–6810.

[42] Zhang, Z., Wang, B., 2021. Fusion of latent categorical prediction and sequential prediction for session-based recommendation. Inf. Sci. 569, 125–137.

[43] Zheng, J., Mai, J., Wen, Y., 2022. Explainable session-based recommendation with meta-path guided instances and self-attention mechanism, in: SIGIR, ACM, Madrid, Spain. pp. 2555–2559.

[44] Zheng, L., Li, Z., Li, J., Li, Z., Gao, J., 2019. Addgraph: Anomaly detection in dynamic graph using attention-based temporal GCN, in: Kraus, S. (Ed.), IJCAI, ijcai.org, Macao, China. pp. 4419–4425.

[45] Zhou, R., Qian, H., Zhang, J., Liu, C., Wan, J., Ren, Y., Xiong, N., Zhao, N., Zhang, S., 2022. Self-attention mechanism enhanced user interests modeling for personalized recommendation services in cyber-physical-social systems. IEEE Trans. Netw. Sci. Eng. 9, 3835–3846.

[46] Zhu, H., Li, X., Zhang, P., Li, G., He, J., Li, H., Gai, K., 2018. Learning tree-based deep model for recommender systems, in: SIGKDD, ACM, London, UK. pp. 1079–1088.

[47] Ziegler, C., McNee, S.M., Konstan, J.A., Lausen, G., 2005. Improving recommendation lists through topic diversification, in: Ellis, A., Hagino, T. (Eds.), WWW, ACM, Chiba, Japan. pp. 22–32.

[48] Zuo, Y., Liu, G., Lin, H., Guo, J., Hu, X., Wu, J., 2018. Embedding temporal network via neighborhood formation, in: SIGKDD, ACM, London, UK. pp. 2857–2866.