MoMENt: Marked Point Processes with Memory-Enhanced Neural Networks for User Activity Modeling

SHERRY SAHEBI, State University of New York at Albany, Albany, USA
MENGFAN YAO, State University of New York at Albany, Albany, USA
SIQIAN ZHAO, State University of New York at Albany, Albany, USA
REZA FEYZI BEHNAGH, State University of New York at Albany, Albany, USA

Marked temporal point process models (MTPPs) aim to model event sequences and event markers (associated features) in continuous time. These models have been applied to various application domains where capturing event dynamics in continuous time is beneficial, such as education systems, social networks, and recommender systems. However, current MTPPs suffer from two major limitations, i.e., inefficient representation of event dynamic's influence on marker distribution and losing fine-grained representation of historical marker distributions in the modeling. Motivated by these limitations, we propose a novel model called Marked Point Processes with Memory-Enhanced Neural Networks (MoMENt) that can capture the bidirectional interrelations between markers and event dynamics while providing fine-grained marker representations. Specifically, MoMENt is constructed of two concurrent networks: Recurrent Activity Updater (RAU) to capture model event dynamics and Memory-Enhanced Marker Updater (MEMU) to represent markers. Both RAU and MEMU components are designed to update each other at every step to model the bidirectional influence of markers and event dynamics. To obtain a fine-grained representation of maker distributions, MEMU is devised with external memories that model detailed marker-level features with latent component vectors. Our extensive experiments on six real-world user interaction datasets demonstrate that MoMENt can accurately represent users' activity dynamics, boosting time, type, and marker predictions, as well as recommendation performance up to 76.5%, 65.6%, 77.2%, and 57.7%, respectively, compared to baseline approaches. Furthermore, our case studies show the effectiveness of MoMENt in providing meaningful and fine-grained interpretations of user-system relations over time, e.g., how user choices influence their future preferences in the recommendation domain.

CCS Concepts: • **Mathematics of computing** → *Stochastic processes*; • **Information systems** → *Recommender systems*; *Spatial-temporal systems*; • **Computing methodologies** → *Neural networks*; *Multi-task learning*;

Additional Key Words and Phrases: Point process, Hawkes process, user activity modeling, sequential model

ACM Reference Format:
Sherry Sahebi, Mengfan Yao, Siqian Zhao, and Reza Feyzi Behnagh. 2024. MoMENt: Marked Point Processes with Memory-Enhanced Neural Networks for User Activity Modeling. *ACM Trans. Knowl. Discov. Data.* 18, 6, Article 155 (April 2024), 32 pages. https://doi.org/10.1145/3649504

This paper is based upon work supported by the National Science Foundation under Grants Numbers 2047500 and 1917949.
Authors' address: S. Sahebi, M. Yao, S. Zhao, and R. Feyzi Behnagh, State University of New York at Albany, 1400 Washington Ave, Albany, New York, USA, 12222; e-mails: ssahebi@albany.edu, myao@albany.edu, szhao2@albany.edu, rfeyzibehnagh@albany.edu.

ACM 1556-4681/2024/04-ART155
https://doi.org/10.1145/3649504

1 INTRODUCTION

Modern online systems such as social media, e-commerce platforms, and online learning systems produce a massive amount of user-system interaction data on a daily basis. Such data contain rich information about user behaviors, such as when they interact (i.e., interaction time), how they interact (i.e., types of interactions such as ordering a product vs. browsing it), and what possible feedback they receive or provide (i.e., markers, such as the rating value they provide to a product). For example, in an e-commerce platform, a user may rate (a type of interaction) a product on Friday night (time) as a five-star product (marker). Likewise, in an online learning system, a student may attempt a quiz (a type of interactive event) on Monday morning (time) and provide a correct answer (marker). Recent research has shown that the timings of these interactive events, particularly their historical intensities, provide meaningful representations of users' true interactive events and hence are important for the prediction of their upcoming behaviors [17, 18].

Marked Temporal Point Process models (MTPPs) [32] are particular types of **Temporal Point Process models (TPPs)** [12] that represent activity timings along with their associated markers. A marked temporal point process is characterized by its activity intensity function, defined as a continuous function of time and markers, conditioned on activity history. This conditional intensity function allows an MTPP to model activity dynamics in continuous time without a need for discretization. Due to their ability to model activity timings and markers in continuous time, MTPPs are beneficial for application domains such as social networks, recommender systems, or education where modeling and predicting user activity timings and feedback (i.e., markers) are essential.

Despite the successful applications of MTPPs in various domains, they still suffer from two major limitations: ineffective representation of the influence of activity dynamics on marker distributions and the lack of fine-grained representation of historical marker distributions, which hinders model interpretation. First, most MTPPs cannot efficiently model the comprehensive interrelationships between marker distribution and activity dynamics. Traditional MTPPs usually specify the empirical conditional intensity function as a function of markers and capture the potential influence of markers on future activities. However, their underlying assumption is that the markers are i.i.d. and invariant to the activity arrival times. This limits the models' ability to predict markers and fully capture the underlying dynamics of the data. Additionally, in these methods, the strict parameterization of the intensity function leads to model misspecification and overfitting. More recently, neural MTPPs have been proposed to resolve the strict parameterization issue in traditional MTPPs by using RNNs to model and define intensity functions [9, 17, 71]. These models can achieve higher capacity compared to traditional MTPPs, by allowing a more flexible representation of the data. However, they still fail to comprehensively capture the complex interrelationships between marker and activity timing dynamics. Recent attempts of MTPPs either model markers and activity timestamps via a shared hidden state in one RNN [17] or as two independent dimensions via two RNNs [9, 71]. The first approach assumes that the markers and the activity timings follow exactly the same dynamics. So, the model cannot distinguish between the possibly different dynamics that these two may have. The second approach leads to the assumption that the marker sequence and the activity timings do not have any influence on each other at each time step and can only change independently. We argue that modeling the interrelations between marker and activity dynamics is crucial for MTPPs in applications with complex relationships between the markers and timings and that the simplified assumptions in current MTPPs are inadequate for such applications. First, user activity dynamics and markers do not necessarily follow the same dynamics. For example, in online education, improvement of student learning performance in an online course does not necessarily come after intense studying activities. In fact, directly associating study intensity and learning performance has been shown to result in contradictory conclusions [30]. Likewise, in an

e-commerce platform, a user's less frequent purchasing behaviors (e.g., purchases of electronics compared with groceries) do not always suggest the user's less interest in those products. These behaviors may be explained by a combination of factors, such as various consumption cycles of different products [26].

As a result, modeling activity dynamics and markers via the same representation may be inadequate for such complicated cases (i.e., the first approach). On the other hand, although activity dynamics and markers may not follow exactly the same dynamics, some interrelationships between markers and activity dynamics may exist. For example, a student's future choice of when to practice a topic (activity dynamics) may depend on how knowledgeable the student is in that topic (observed by student performance as markers). Likewise, their current knowledge also depends on their historical practicing behavior, such as how (e.g., which learning materials they used) and when they practiced. As a result, not modeling the relationship between activity dynamics and markers may reduce the model's descriptive power of user interaction data (i.e., the second approach). As the second limitation, modern neural MTPPs model activity histories as abstract state vectors in RNNs. This representation of history can result in losing fine-grained activity-level features in RNN updates and diluting the marker and activity dynamic interpretations. This is particularly deficient for application domains, such as recommendation and education, where understanding and interpreting user behavior is an essential task. Combined with the interrelationship problem mentioned above, the interpretability problem will be even more complicated. In that case, the critical understanding of how marker and activity dynamics interrelate will be compromised, thus making it even harder to explain the data dynamics.

To address the above limitations, we propose a novel neural MTPP model: **Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt)** that captures and represents the bidirectional relations between markers and activities while revealing the detailed patterns in their dynamics. To this end, we model activities and their markers as two concurrent networks in MoMENt updating each other with historical activity dynamics at each step. Specifically, for marker dynamics, we rely on a **Memory-Augmented Neural Network (MANN)** [23] that can provide structured memory slots to achieve more stable and stronger performance than standard RNNs. We propose **Recurrent Activity Updater (RAU)** and **Memory-Enhanced Marker Updater (MEMU)** in MoMENt that respectively model activity dynamics and markers, and capture the influence of markers on activities and vice versa (addressing the first limitation). The RAU component extends a **Long short-term memory (LSTM)** network to model activity dynamics while considering the markers. Furthermore, the MEMU component in MoMENt utilizes external memories to obtain a fine-grained representation of maker distributions (addressing the interpretability limitation in the state of the art).

We formulate MoMENt based on the problem of user activity modeling in continuous time. The prediction layer in MoMENt is designed to predict user activity time, type, and outcome (markers) simultaneously. For activity time modeling in the prediction layer, we propose a new activity distribution model that has the flexibility of neural representations in the deep Hawkes models while keeping the interpretability of the traditional Hawkes intensity functions. Unlike the traditional Hawkes processes, MoMENt's formulation of intensity function allows for all positive, negative, and neutral types of internal influence from past activities to future ones. Additionally, unlike the more recent neural Hawkes process models, MoMENt's intensity function is interpretable.

Our extensive experiments on six real-world datasets show consistent improvements in MoMENt over the state-of-the-art approaches, demonstrating the power of MoMENt in accurately describing and predicting user interaction timings and types. Specifically, the performance of MoMENt compared to 16 baseline models and their variations in two application domains, demonstrate significant improvements in all classification, prediction, and ranking tasks. This includes

up to 77.2% gain in AUC in student marker prediction, 65.6% gain in ACC in student activity type prediction, 76.5% improvement in RMSE in student activity time prediction, and 57.7% average gain in HR@10 in the recommender system's next item prediction. Our ablation studies, removing the time and type inputs from the model, demonstrate the significance of MoMENt's contribution in modeling the activity to marker influence, in addition to the marker to activity influence. Furthermore, our case studies and interpretation analyses show the effectiveness of MoMENt in representing user-system relations over time in a fine-grained manner to demonstrate meaningful associations with the temporal patterns of user interactions.

To summarize, the main contributions of this work are:

— Introducing a novel neural MTPP model, MoMENt, that captures and represents the bidirectional relations between markers and activities;
— Proposing a new activity distribution model with multiple influence types that has the interpretability of the traditional Hawkes intensity functions while benefiting from the flexibility of neural representations in the deep Hawkes models;
— Introducing MoMENt's prediction layer that can simultaneously predict activity time, type, and marker according to the historical activity information;
— Proposing the MEMU component, a novel MANN-based architecture that can model fine-grained activity and marker-level features while including activity dynamics information;
— Presenting the RAU component, an LSTM-based architecture to model activity dynamics, given fine-grained marker information;
— Thorough experiments to study the performance of MoMENt, extensive ablation studies, and analysis and interoperability of MoMENt.

In the following, we first provide a brief overview of related work (Section 2) and backgrounds (Section 3). Then, we introduce our problem formulation (Section 4) and the MoMENt model (Section 5), including the MEMU and RAU units. Finally, in Section 6, we present our experiments on six real-world datasets in two application domains. We evaluate MoMENt by studying its predictive performance, ablation study, and interpretation and analysis of its discovered interrelationships.

2 RELATED WORK

Temporal Point Processes are stochastic processes that model events characterized by their timings. Unlike discrete models such as time series or conventional sequential models that rely on the discretizations of time, temporal point process models aim to describe the dynamic of events in continuous time. For this reason, they are advantageous for many real-world applications where understanding temporal patterns of the data and predicting future event dynamics are important, such as financial applications (e.g., [2, 43]), education (e.g., [38, 73, 74]), or information diffusion modeling (e.g., [5, 19, 20]).

Marked temporal point processes, as a special family of point process models, characterize activities not only by their timings but also their associated features (i.e., markers). In traditional approaches, marked temporal point process models (MTPPs) typically use markers to parameterize the activity intensity function, assuming that activity intensity is affected by the marker distributions. For example, Mishra et al. [46] assume that an X (formerly known as Twitter) user's social influence decides the popularity of their tweets, and parameterized the intensity of a user's tweets being retweeted as a function of the user's number of followers (marker) as a proxy of their influence (i.e., markers). A similar approach was proposed by Zhao et al. [79], where the modeling of the number of followers and timestamps are realized by the Cox process. This method of parameterization typically necessitates domain knowledge and preliminary research, because the properties of markers and how they define activity intensities vary depending on the applications and datasets.

Recently, neural TPPs have been proposed that use RNNs to model time dependencies and avoid misspecification that most traditional point processes suffer, e.g., intensity functions characterize the activity dynamics following a rigid form [17, 70, 71, 78, 84]. Apart from providing a more flexible formulation, another important contribution of neural TPPs is the efficient integration of marker modeling with modeling sequence activities and their times. For example, Du et al. [17] propose to model a point process using RNN and use the hidden state at each time step to represent the intensity function of both marker embedding and activity embedding, which captures the effect of both historical markers and activity timestamps. However, this representation of intensity is not sufficient when the underlying dynamics of markers and activity times are not similar. In more recent literature on neural marked point process models, a more common approach is used by summing up the representations of markers and activity timestamps modeled by two independent RNNs. As a result, the interrelationship between markers and activity timestamps and how their historical dynamics affect each other's future is not directly modeled. For example, Xiao et al. [71] use two LSTMs that independently model the sequence of markers and the sequence of activity timings. An improvement of this model is then proposed by Xiao et al. [70] where self-attention was added to increase the model's interpretability of time dependencies. Similarly, Choi et al. [9] add an attention mechanism on top of two independent RNNs. Furthermore, as another limitation, the above approaches model markers via standard RNNs or LSTMs that forcefully represent the historical information as an abstract vector, which is not only hard to interpret but also compromises the fine-grained representation of markers. Our proposed work is the first to model the bidirectional influence between markers and activities, propose an interpretable intensity function to model positive and negative influence types while maintaining the flexibility of deep representations, capture the fine-grained marker dynamic distributions, and predict event time, type, and markers simultaneously.

User Activity Modeling in online systems has become one of the most important tasks for understanding user preferences and intentions. For example, in modern online education systems such as online teaching and tutoring platforms, modeling students' learning behaviors and quantifying students' knowledge as they learn is an essential task [22, 50, 52, 64, 76]. In recent developments of student learning and knowledge modeling, for example, **Long-short term memory (LSTM)** [52], **memory-augmented neural network (MANN)** [76], and encoder-decoder [22] have been employed to model student evolving knowledge using their ordered learning activities as input. More recently, more efforts have been made to take activity timestamps or the duration between activities into account, to better represent the temporal aspects of student learning [8, 22, 56, 61]. For example, Wang et al. [61] use Hawkes process's intensity function to model students' learning timings to capture the relatedness between students' skill sets. However, none of these methods predicts students' future activity time or takes into account multiple types of learning materials, making them insufficient for accurately representing students' learning processes [48, 51].

Another application domain that has made extensive use of user activity modeling is *recommender systems*, with the goal of suggesting useful items to users by inferring their intention to consume or like a given item. Due to the advantage of modeling user interaction dynamics over static methods such as collaborative-filtering-based methods [3, 34, 37, 53], sequential modeling has become one of the most popular approaches for this task. Nonetheless, conventional sequential methods mostly consider only the orders of users' activities without taking into account their timings, with the underlying assumption that users' behaviors and preferences are time-invariant and linear to their orders [16, 40, 68, 75, 82]. More recently, some time-aware sequential models have been proposed to tackle this limitation [4, 7, 66, 67, 69, 83]. For example, Wang et al. [62] propose to combine the Hawkes process with collaborative filtering with a focus on the repeated

consumption of each item. With a customized kernel function, more recently, Wang et al. [63] propose Chorus to model the dynamics of item relations by learning graph representations of items over time. Wang et al. [60] further propose an improvement over Chorus by introducing the Fourier transform, which is subsequently leveraged by the self-attention mechanism to obtain a dynamic historical representation of user activities. Nevertheless, these methods lack the bidirectional influence and the interpretability of users' evolving preferences or interests in items over time that is modeled by our work.

3 TEMPORAL POINT PROCESS BACKGROUND

A **temporal point process (TPP)** can be defined as a set of points (e.g., events) that fall randomly in time. This temporal representation is ideal for modeling a collection of events, such as user activities, that are characterized by their arrival timestamps. Temporal point process models are methods that describe and model these point processes that can be characterized by the conditional density function f, as a function of time t, given the historical observations of event timings $\mathcal{H}_t$. The joint density function for the realization of historical observations $\{x^1, ..., x^K\}$, can be obtained as:

$$f(x^1, ..., x^K) = \prod_{\tau=1}^{K} f(x^\tau | \mathcal{H}_{\tau-1}). \tag{1}$$

Where x^τ can take the form of activity time for the τ^{th} activity for conventional TPPs, or the representation of both activity time and available relevant markers. For ease of computation and a better interpretation, it is common to use an alternative conditional event intensity function $\lambda(t|\mathcal{H}_{x^n})$, usually simplified as $\lambda^*(t)$ in the literature [11]. $\lambda(t|\mathcal{H}_{x^n})$ represents the conditional intensity distribution of activity arrivals conditioned on the history $\mathcal{H}_{x^n} = \{x^1, ..., x^n\}$ up to time t where x^n is the most recent event before t. It is a function of $f^(t|\mathcal{H}_{x^n})$ and its corresponding cumulative distribution function, $F(t|\mathcal{H}_{x^n})$:

$$\lambda^*(t) = \frac{f(t|\mathcal{H}_{x^n})}{1 - F(t|\mathcal{H}_{x^n})} = \frac{f(t|\mathcal{H}_{x^n})}{1 - \int_{x^n}^{t} f(s|\mathcal{H}_{x^n}) ds}. \tag{2}$$

From the above, we see that an intensity function is the core of TPPs that defines the activity dynamics in continuous time. In the following, we provide an overview of some of the most representative TPPs characterized by their intensity functions in the literature.

- **Poisson Process** is a classic member of the point process family, which can be categorized into homogeneous Poisson processes and non-homogeneous Poisson processes. The former has a simplistic assumption that activities arrive with a constant rate μ, i.e., $\lambda(t) = \mu \geq 0$. On the other hand, with a more general assumption, a non-homogenous process model allows the rate of activity arrivals to be a function of time instead of a constant number, i.e., $\lambda(t) = \mu(t)$, usually with its intensity function customized to the specific applications [28, 31, 39, 55].
- **Hawkes Process** or self-exciting point process is the most widely-adopted point process family, within which activities are assumed to be "self-exciting", meaning that historical activities have a triggering or "exciting" effect on the future ones. These processes are characterized by an intensity function $\lambda(t) = \mu + \sum_{t_\tau < t} \phi(t - t_\tau)$, where μ is the base rate which describes the activities that naturally arrive as a result of external factors, and ϕ is a kernel function that describes the total effects of previous activities on activity arrivals at time t, summed over the entire history. An important equivalent view of Hawkes processes is the branching structure [25], which divides activities into the concepts of "immigrants" and "offsprings." In this interpretation, "immigrants" represent the activities that are externally triggered by the environment (characterized by μ), and "offsprings" represent the activities that

are internally self-excited by immigrants (characterized by ϕ). Due to this assumption's fitness to many real-world scenarios, such as in social media, Hawkes process models with different intensity functions have been proposed and applied to many practical machine learning tasks. Particularly, different kernel functions ϕ, such as exponential (e.g., [18]), Gaussian (e.g., [72]), and power-law (e.g., [46]) have been proposed for different applications.

— **Self-correcting Process**, unlike Hawkes processes, assumes that historical activities have an inhibiting effect on future ones. These processes are characterized by an intensity function $\lambda(t) = \exp(\mu t - \sum_{t_\tau < t} \alpha)$, where μ and α are positive numbers. From this function we see that when an activity arrives, the intensity is divided by $\exp(\alpha) > 1$, which describes the scenarios where the arrival of a new activity decreases the chances of future arrivals of activities as if the process is self-correcting itself to its regularity. With this assumption, self-correcting process models are most popularly used to describe well-dispersed activity occurrences, such as the modeling of earthquakes after aftershocks [49, 54].

— **Recurrent TPPs modeled by deep networks and RNNs** have been proposed more recently to obtain a more flexible formulation of the intensity function. Hidden representations of the recurrent neural networks are used to describe the time dependencies (e.g., self-exciting) and to represent the intensity of the process. For example, Du et al. [17] were the first to propose to use RNN to both capture past and current activity dependencies via the following intensity function: $\lambda(t) = \exp(\mathbf{v}^{t\top} \cdot \mathbf{h}_j + w^t(t - t_j) + b^t)$, where the three terms respectively represent past influence, current influence, and base intensity. Since such intensity function formulations via RNNs are not tailored to a specific scenario or application, they are easier to be adopted from one domain to another [17, 44, 70]. More recently, motivated by transformers [59], the self-attention framework has been used to characterize the intensity function [77, 78, 84]. Self-attention captures different weights of past activities in determining the future activity intensity. Furthermore, more complex structures, such as recurrent graph networks [6, 13], sequential variational autoencoders [15], and convolutional neural networks [81], have been used to model interactions or relationships between various events in learning event dynamics.

Nevertheless, current MTPPs do not model the bidirectional influence between markers and activities, do not include interpretable intensity functions to model positive and negative influence types while maintaining the flexibility of deep representations, do not capture the fine-grained marker dynamic distributions, and do not predict time, type, and markers simultaneously. We address all of these shortcomings in this paper.

4 PROBLEM FORMULATION

We focus on the problem of user activity modeling, considering its importance and its wide range of applications, such as in recommender systems [38, 62, 65] and student sequence modeling [9, 73, 74]. Suppose we are given N users and their interactions with Q items. The collection of all users' activities on items can be represented as $\mathcal{S} = \{S^1, ..., S^N\}$, with S^i representing user i's sequence of activities. Also, suppose that user i's j-th activity is associated with its timestamp t^i_j, its type y^i_j, item-level attributes, such as item id $q^{id}{}^i_j$ or item tag $q^f{}^i_j$, as well as the observed user-item interaction outcome r^i_j, i.e., marker. In this way, a user sequence that contains K activities can be represented as a collection of 5-tuples: $\{(t_j, y_j, q^{id}_j, q^f_j, r_j) | j = 1, ..., K\}$ [1]. Note that some of these observations may be missing for some activities, such as missing q^f, or unobserved r, which will

[1]We omit user index i for presentation simplicity.

be padded as −1. Our goal is to predict users' future activity times, types, and outcomes, given their interaction history, by modeling the complex trends in the activity and interaction outcome dynamics and the associations between them.

Under this formulation, we consider the following four assumptions. First, we consider the historical influence of past activities on future activities (i.e., *activity2activity* influence). This assumption corresponds to the time dependency assumption that has been used in many point process families, such as in Hawkes processes, e.g., a user's historical activities can trigger follow-up activities. For example, in an online course, a student watching a video lecture may lead to follow-up activities such as trying out related quiz questions. Similarly, a user who bought a product (e.g., nail polish) online may consequently buy more related products (e.g., nail polish remover). Second, we assume the influence of past markers on future activities (i.e., *marker2activity* influence), which is a common assumption that has been used in MTPPs, where the markers are assumed to be predictive of the future activity intensity. For example, a student who has failed a quiz may follow up with activities such as reviewing related lectures, and a user who has had a satisfactory purchase from a website in the past may interact with it more often in the future.

Third, to better describe the complex dynamics presented in the data, instead of assuming that markers are i.i.d., we assume the historical influence of markers on the future marker distribution (i.e., *marker2marker* influence). In other words, the marker distribution can change over time, as a reflection of the evolving user-system relationship and the markers' historical distributions. For example, students' past grades, as a reflection of their evolving knowledge of course concepts, can be predictive of their future grades. Or, user satisfaction, as a reflection of the user-system relationship can evolve according to the user's historical satisfaction with the system.

Finally, we assume the influence of historical activities on the markers' future distribution (i.e., *activity2marker* influence). For example, students' historical learning activities and their study pace can affect their future grades. Likewise, a series of intense browsing activities in the system may conclude with the user's satisfactory purchase. The first two assumptions above have been widely used in the literature on point process modeling. However, the last two assumptions, namely *activity2marker* and *marker2marker* influences, have been largely overlooked in the literature. We argue that these two assumptions are essential in the simultaneous modeling of user activities and their markers, and we build MoMENt to include these assumptions in addition to the widely used ones.

5 MARKED POINT PROCESSES VIA MEMORY-ENHANCED NEURAL NETWORKS (MoMENt)

In this section, we formally introduce our model, Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt). An overview of MoMENt is presented in Figure 1. For each sequence step, it includes an input layer, a **Recurrent Activity Updater (RAU)**, a Memory-Enhanced Marker Updater (MEMU), and a prediction layer.

At step j, in the input layer, the 5-tuple $(t_j, y_j, q_j^{id}, q_j^f, r_j)$ is used as input and is embedded to obtain activity dynamic embedding $\mathbf{e}_{\mathbf{a}j}$ and marker embedding $\mathbf{e}_{\mathbf{m}j}$ (Section 5.1). To capture the complex dynamics of user activities and markers, we model them via two modules. Activity dynamic embedding is modeled by Recurrent Activity Updater (RAU) to capture the historical influence of activity arrivals and markers on the future activity dynamics i.e., activity2activity (full yellow arrow) and marker2activity (dashed green arrow) (Section 5.3). In parallel, Memory-Enhanced Marker Updater (MEMU) is proposed to model the influences of historical markers and activity dynamics on the future marker distributions, i.e., marker2marker (dash-dotted red arrow) and activity2marker (dotted purple arrow). Inspired by Memory-Augmented Neural Networks (MANNs),

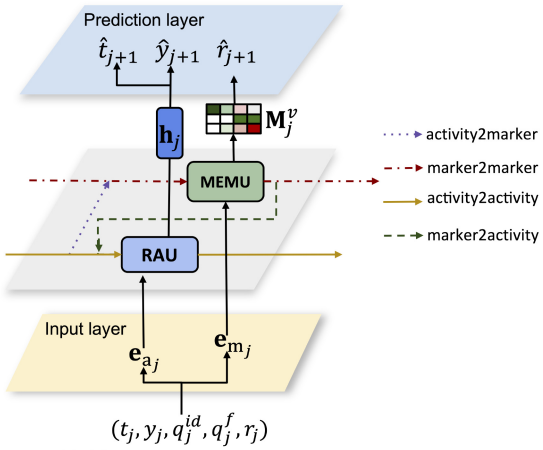

Fig. 1. Overview of the proposed model MoMENt that consists of input layer, Recurrent Activity Updater (RAU), Memory-Enhanced Marker Updater (MEMU), and prediction layer. Four influence types, namely activity2marker, marker2marker, activity2activity, and marker2activity, are captured during RAU and MEMU updates.

MEMU utilizes external memory matrices to represent each user's evolving relationship with all items, to provide a fine-grained representation of marker distribution (Section 5.2).

Note that since MoMENt has separate RAU and MEMU components, the activities and markers do not need to follow the exact same dynamics. At the same time, RAU and MEMU components are not modeled independently. Rather, the activity2marker and marker2activity connections allow the two components to communicate and coordinate. Finally, the obtained hidden representations of RAU and MEMU are used to respectively predict the next activity time $\hat{t}_{j+1}$ and type $\hat{y}_{j+1}$, as well as activity marker $\hat{r}_{j+1}$ in the prediction layer (Section 5.4). The details of each model component are provided below.

5.1 Input Layer

For activity type y_j, we first represent it as a one-hot vector $\mathbf{y}_j$. As point processes can have many types, we optionally apply a linear transformation on $\mathbf{y}_j$ to obtain a more compact activity type representation $\mathbf{e}_{\mathrm{y}_j} = \mathbf{W}_{\mathrm{y}}\mathbf{y}_j$. Furthermore, we represent the timing of activity x_j as the inter-arrival time between the j^{th} and $(j-1)^{th}$ activities. That is, if the j^{th} activity takes place at time t_j, we set $x_j = t_j - t_{j-1}$. This is in accordance with the definition of point processes, which is based on inter-arrival times. Finally, we use $\mathbf{e}_{\mathrm{a}j} = [x_j; \mathbf{e}_{\mathrm{y}_j}]$ to represent activity dynamic embedding. To encode item id q_j^{id}, we generate its one-hot representation $\mathbf{q}^{\mathrm{id}}{}_j$. For categorical item feature q_j^f, we also generate its one-hot representation $\mathbf{q}^{\mathrm{f}}{}_j$ if it is singular, otherwise we apply a multi-label Binarizer that binarizes multi labels into 1 or 0. For categorical marker r_j, we similarly first obtain its one-hot representation $\mathbf{r}_j$. As each user-item interaction outcome depends on the item, we concatenate item-level features with the marker to obtain the marker embedding $\mathbf{e}_{\mathrm{r}j} = [\mathbf{q}^{\mathrm{id}}{}_j, \mathbf{q}^{\mathrm{f}}{}_j, \mathbf{r}_{\mathrm{j}}]$. To also encode time information and to provide more interpretability, we follow Li et al. [42] to use a time mask in obtaining the final marker representation $\mathbf{e}_{\mathbf{m}}$. This mask is used in an element-wise product with marker representation $\mathbf{e}_{\mathbf{r}}$, so that less important historical activities and their influence will be masked. To do this, we first create a time context vector c_x for the time representation x by computing $c_x = \sigma(\mathbf{W}_{\mathbf{m}}\phi(x) + \mathbf{b}_{\mathbf{m}})$. ϕ is a nonlinear transformation of $log(x)$, implemented

as a feedforward neural network; and $\mathbf{W_m}$ and $\mathbf{b_m}$ are weight matrix and bias vectors to linearly transform $\phi(x)$. Then, by performing an element-wise product of c_x and $\mathbf{e_r}$, we create the final time-masked marker embedding $\mathbf{e_m}$. Eventually, we have $\mathbf{e_m} = \mathbf{e_r} \odot c_x = \mathbf{e_r} \odot \sigma(\mathbf{W_m}\phi(x) + \mathbf{b_m})$.

5.2 Memory-Enhanced Marker Updater (MEMU)

The proposed Memory-Enhanced Marker Updater aims to increase model interpretability while capturing two important dependencies that have been overlooked by the literature, namely the marker2marker influence, and activity2marker influence. Conventionally, markers and activity timings are modeled via standard RNNs or LSTMs. Such methods express the history of a given sequence as an abstract dense vector, which compromises fine-grained features of markers that relate to user-system relationship, and also is hard to interpret. Memory-Augmented Neural Networks (MANNs), on the other hand, are forms of neural networks that have shown to achieve more effective performances than standard RNNs while providing fine-grained and meaningful interpretations [23], especially in application domains with complex marker features and associations [64, 76]. MANNs achieve this by using external memory matrices to enable read and write operations [23, 57] that provide structured memory slots leading to local state storage and transactions. However, MANNs have not been considered for MTPPs before.

Motivated by the limitation of using RNNs to model marked point processes, we propose an adaptation of the Key-Value Memory Network [45] to enhance the representation of markers during RNN updates. A Key-Value Memory Network is a special type of MANN that consists of a key matrix and a value matrix that respectively store static keys and dynamic values over time. We propose to use the key matrix $\mathbf{M}^k \in \mathbb{R}^{C \times d_k}$ to model item-level features, supposing all items have C latent components and each component is measured by d_k latent variables. The key matrix is set to be static, assuming that item features do not change over time. Furthermore, we use the value matrix $\mathbf{M}_j^v \in \mathbb{R}^{C \times d_v}$ to model the user-item relationship with all C item components at step j, described by d_v latent variables. $\mathbf{M}_j^v$ is set to be dynamic to capture the user's evolving relationship with the system. The above key-value structure to model the markers offers extra modeling flexibility because of the evolving value matrix vs. the static key matrix. It also provides a fine-grained explanation of the data based on the mapping interpretation from the value to the key (e.g., student knowledge of all concepts as an explanation of learning outcomes).

We define MEMU as $\mathbf{M}_j^v = \text{MEMU}(\mathbf{e}_{\mathbf{m}j}, \mathbf{h}_{j-1}, \mathbf{M}^k, \mathbf{M}_{j-1}^v)$, where the current state of marker $\mathbf{M}_j^v$ is updated by the current marker embedding $\mathbf{e}_{\mathbf{m}j}$, the previous hidden state of activity dynamics $\mathbf{h}_{j-1}$ (will be introduced in Section 5.3, which presents RAU), the static key matrix, and the previous state of marker $\mathbf{M}_{j-1}^v$. An illustration of MEMU is given in Figure 2.

Specifically, MEMU is defined as follows:

$$\mathbf{e}_j = [\mathbf{h}_{j-1}; \mathbf{e}_{\mathbf{m}j}], \tag{3}$$

$$\mathbf{q}_j = \mathbf{W_q}\mathbf{e}_j, \tag{4}$$

$$\mathbf{a}_j = \text{softmax}\left(\frac{\mathbf{M}^k\mathbf{q}_j}{\sqrt{d_c}}\right), \tag{5}$$

$$\mathbf{V}_j = \mathbf{W}_V\mathbf{e}_j\mathbf{u}_V^\top, \tag{6}$$

$$\mathbf{s}_j = \left(\mathbf{a}_j^\top\mathbf{V_j}\right)^\top, \tag{7}$$

$$\tilde{\mathbf{M}^v}_j = \tanh\left(\mathbf{W_M}\mathbf{M}_{j-1}^v + \mathbf{s}_j{\mathbf{u_M}}^\top + \mathbf{b_M}\right), \tag{8}$$

$$\mathbf{Z}_j = \sigma\left(\mathbf{W_Z}\tilde{\mathbf{M}}_j^v + \mathbf{s}_j{\mathbf{u_Z}}^\top + \mathbf{b_Z}\right), \tag{9}$$

$$\mathbf{M}_j^v = \mathbf{Z}_j \odot \mathbf{M}_{j-1}^v + (1 - \mathbf{Z}_j) \odot \tilde{\mathbf{M}}_j^v. \tag{10}$$

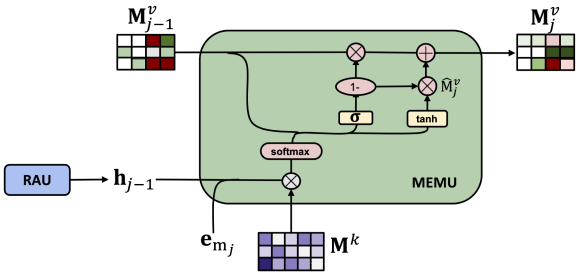

Fig. 2. An illustration of MEMU.

In Equation (3), we first concatenate the current marker embedding $\mathbf{e}_{\mathrm{m}_j}$ with the previous hidden state $\mathbf{h}_{j-1}$ from RAU to obtain $\mathbf{e}_j$. The resulting embedding is multiplied with the embedding matrix $\mathbf{W}_q$ to obtain an embedding vector $\mathbf{q}_j$ of length d_k, via Equation (4). A scaled dot-product [59] is applied to the resulting embedding $\mathbf{q}_j$ and key matrix $\mathbf{M}^k$, followed by a softmax via Equation (5), to obtain a weight vector $\mathbf{a}_j$ of length C. The obtained weight $\mathbf{a}_j$ can be interpreted as the correlation between the current item and all item components, considering the user's historical interactions. Next, by using Equation (6), the original embedding vector $\mathbf{e}_j$ is embedded in matrix $\mathbf{V}_j \in \mathbb{R}^{C\times d_v}$, which can be interpreted as a fine-grained representation of the interaction outcome (i.e., marker) of all C components. This representation is finally multiplied with the weight vector $\mathbf{a}_j$ via Equation (7) to get the weighted fine-grained outcome in terms of C components, considering historical influences of interactions. The resulting representation of the outcome is then used to update the user-item relationship in the next state, i.e., $\mathbf{M}_j^v$, via Equations (8)–(10). These equations follow a routine similar to GRU [10], to allow "adding" and "erasing" the historical effects of both user interaction dynamics (i.e., $\mathbf{s}_j$ computed based on $\mathbf{h}$) and user relationship with items (i.e., $\mathbf{M}_j^v$). However, note that, unlike prior versions of MANNs, like [45, 76], MEMU keeps track of the dynamics for both marker sequence and the previous hidden state of activity dynamics from the RAU component that will be introduced shortly. As a result, MEMU's formulation is different from the previously introduced MANN-like structure in the literature.

5.3 Recurrent Activity Updater (RAU)

Now we introduce the framework of our Recurrent Activity Updater (RAU), defined as $(\mathbf{h}_j,\mathbf{c}_j) = \mathrm{RAU}(\mathbf{e}_{\mathrm{a}j}, \mathbf{h}_{j-1}, \mathbf{c}_{j-1}, \mathbf{M}_j^v)$. We can see that the output contains the current hidden state $\mathbf{h}_j$, and the cell state $\mathbf{c}_j$, which respectively have the same interpretations as they have in Long Short Term Memory (LSTM) [29]. The input data on the other hand contains the current activity embedding $\mathbf{e}_{\mathrm{a}j}$, the previous hidden state $\mathbf{h}_{j-1}$, the previous cell state $\mathbf{c}_{j-1}$, and finally the current state of the value matrix $\mathbf{M}_j^v$ from the MEMU component. The core idea of this updater is to adopt the LSTM framework to update $\mathbf{h}_j$ taking into account the effect of historical activities in terms of their timings and types (i.e., $\mathbf{h}_{j-1}$ and $\mathbf{c}_{j-1}$) as well as the historical influence of markers (i.e., $\mathbf{M}_{j-1}^v$) on future activities. An illustration of RAU is given in Figure 3. More specifically, RAU is defined via the following equations:

$$\mathbf{i}_j = \sigma\big(\mathbf{W}_{\mathrm{i}}\mathbf{e}_{\mathrm{a}j} + \mathbf{U}_{\mathrm{i}}\mathbf{h}_{j-1} + \mathbf{V}_{\mathrm{i}}\mathbf{c}_{j-1} + \mathbf{P}_{\mathrm{i}}\mathbf{M}_j^v\mathbf{y}_{\mathrm{i}} + \mathbf{b}_{\mathrm{i}}\big), \tag{11}$$

$$\mathbf{f}_j = \sigma\big(\mathbf{W}_{\mathrm{f}}\mathbf{e}_{\mathrm{a}j} + \mathbf{U}_{\mathrm{f}}\mathbf{h}_{j-1} + \mathbf{V}_{\mathrm{f}}\mathbf{c}_{j-1} + \mathbf{P}_{\mathrm{f}}\mathbf{M}_j^v\mathbf{y}_{\mathrm{f}} + \mathbf{b}_{\mathrm{f}}\big), \tag{12}$$

$$\mathbf{c}_j = \mathbf{f}_j \odot \mathbf{c}_{j-1} + \mathbf{i}_j \odot \tanh\big(\tag{13}$$

$$\mathbf{W}_{\mathrm{c}}\mathbf{e}_{\mathrm{a}j} + \mathbf{U}_{\mathrm{c}}\mathbf{h}_{j-1} + \mathbf{V}_{\mathrm{c}}\mathbf{c}_{j-1} + \mathbf{P}_{\mathrm{c}}\mathbf{M}_j^v\mathbf{y}_{\mathrm{c}} + \mathbf{b}_{\mathrm{c}}\big),$$

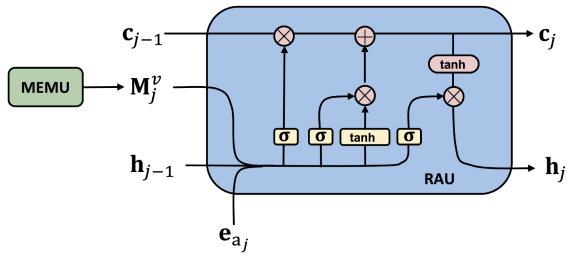

Fig. 3. An illustration of RAU.

$$\mathbf{o}_j = \sigma\big(\mathbf{W}_\mathbf{o}\mathbf{e}_{a_j} + \mathbf{U}_\mathbf{o}\mathbf{h}_{j-1} + \mathbf{V}_\mathbf{o}\mathbf{c}_{j-1} + \mathbf{P}_\mathbf{o}\mathbf{M}_j^v\mathbf{y}_\mathbf{o} + \mathbf{b}_\mathbf{o}\big), \tag{14}$$

$$\mathbf{h}_j = \mathbf{o}_j \odot \tanh(\mathbf{c}_j). \tag{15}$$

As we can see in the above equations, similar to LSTM, considering the previous influence of activities (i.e., $\mathbf{h}_{j-1}$) and markers (i.e., $\mathbf{M}_j^v$), RAU updates its input state $\mathbf{i}$, forget state $\mathbf{f}$, cell state $\mathbf{c}_j$, output $\mathbf{o}$, and hidden state $\mathbf{h}$ respectively in Equations (11)–(15). We see that during computation of the forget gate and cell state, the influence further past to the future activities are allowed to be forgotten, while newer information is allowed to be added. As a result, by using the defined framework, RAU is able to model the long-term dependencies of historical activities and markers.

RAU's design has a straightforward structure compared to MAMU, which should model the complex item, marker, and user-item relationship information. Modeling activity dynamics in RAU is less complicated, as its input $\mathbf{e}_{a_j}$ is less complex than MAMU, and RAU's abstract and coarse-grained dynamic state would suffice for perfectly capturing such dynamics.

5.4 Prediction Layer

The prediction layer consists of three components which respectively predict activity time, type, and outcome for users.

5.4.1 Activity Time Prediction. In this task, we aim to predict when a user will have her/his next activity. This is achieved via two steps: activity time distribution modeling, and predicting inter-arrival time.

Activity time distribution modeling. By definition, a point process can be characterized by its intensity function of time λ^* conditioning on the history of activities. Naturally, since $\mathbf{h}_j$ represents the historical influence of activity dynamics, the intensity function of user interaction sequence can be represented as a function of $\mathbf{h}_\mathbf{j}$. More specifically, we propose to use the following function to define λ^*:

$$\lambda^*(t) = \text{relu}\{b_b + \alpha(\mathbf{h}_j)\exp\big(-\beta(\mathbf{h}_j)(t - x_j)\big)\}, \text{ where} \tag{16}$$

$$\beta(\mathbf{h}_j) = \big(\mathbf{w}_\beta^\top \mathbf{h}_j\big)_+, \tag{17}$$

$$\alpha(\mathbf{h}_j) = \tanh\big(\mathbf{w}_\alpha^\top \mathbf{h}_j\big). \tag{18}$$

In Equation (16), b_b can be interpreted as the base rate, or the number of activities that naturally arrive due to external effects; $\alpha(\mathbf{h}_j)$ defined in Equation (18) describes the number of activities at t that arrive due to the influence of the previous activities (e.g., the past activities triggering the future ones).

For the activation function, we choose tanh due to its flexibility in transforming the values to the range of $[-1, +1]$, to capture various possible time dependencies, such as self-triggering (i.e., $\alpha > 0$), self-inhibiting (i.e., $\alpha < 0$), or memoryless (i.e., $\alpha = 0$) in each time step. Finally, β, as

defined in Equation (17), explains how fast the influence of the past activities decays over time, described by an exponential function in Equation (16). To model this decay rate, we first linearly transform $\mathbf{h}$ by multiplying it with a weight vector $\mathbf{W}_\beta$, and then project it to the positive space, denoted as $(.)_+$, assuming that the past influence decreases over time.

This intensity function has a similar structure to the intensity function of the most widely used point processes, i.e., Hawkes processes, with the following improvements: 1. Unlike traditional Hawkes processes that restrict α to be positive (i.e., strictly triggering effects) and invariant to time, $\alpha(\mathbf{h}_j)$ defined above allows the internal influence from past activities to be negative, or 0. This formulation provides a more flexible representation of complex time dependencies, such as self-inhibiting or Poisson-like dependencies; 2. Compared with the more recent Hawkes process models with neural representations that are hard to interpret, this intensity function has the interpretation of several key phenomenons described in traditional point processes, such as the decaying internal influence, that are characterized by α and β.

Inter-arrival time modeling. Inspired by the sampling strategy used in traditional TPPs, we predict the inter-arrival time between the current activity and the future one as a way to estimate the next activity timing $\hat{t}_{j+1}$. This routine is realized as follows:

$$f^*(t) = \lambda^*(t) \exp\left(-\int_{x_j}^{t} \lambda^*(\tau) d\tau\right), \tag{19}$$

$$g_{j+1} = \tanh\left(\mathbf{w}_\mathbf{g}^\top \log\left(f^*(t_j)\right) + \mathbf{b}_\mathbf{g}\right), \tag{20}$$

$$\hat{x}_{j+1} = \text{relu}\left(\mathbf{w}_\mathbf{t}^\top g_{j+1} + \mathbf{b}_\mathbf{t}\right), \tag{21}$$

$$\hat{t}_{j+1} = t_j + \hat{x}_{j+1}. \tag{22}$$

More specifically, we first obtain the *conditional arrival distribution* $f^*(t)$ that describes the distribution of inter-arrival times via Equation (19) that describes its relationship to the intensity function λ^*. Once we have $f^*(t)$, we can predict inter-arrival time $\hat{x}_{j+1}$ using Equations (20)–(21). We feed the log density $\log(f^*(t_j))$ to a fully connected network and choose relu as the last activation function to guarantee the idle time to be non-negative. The prediction of the next activity time can be then computed via Equation (22), which sums up the current activity time t_j and the predicted next inter-arrival time $\hat{x}_{j+1}$.

5.4.2 Activity Type Prediction. Our goal in this task is to predict the type of $(j+1)^{th}$ activity $\hat{y}_{j+1}$, before observing the activity information, such as item id or tag. For that, we compute the probability of y_{j+1} being of type $y_d \in \mathcal{Y}$ via a softmax function defined as below:

$$P(y_{j+1} = y_d | \mathbf{h}_j) = \frac{\exp(\mathbf{w}_\mathbf{y}{}^\top \mathbf{h}_j + \mathbf{b}_\mathbf{y})}{\sum_{y_d \in |\mathcal{Y}|} \exp(\mathbf{w}_\mathbf{y}{}^\top \mathbf{h}_j + \mathbf{b}_\mathbf{y})}. \tag{23}$$

Then, the $y_d \in \mathcal{Y}$ that gives the highest $P(y_{j+1})$ will be used as the predicted $\hat{y}_{j+1}$

5.4.3 Marker Prediction. In this task, we aim to predict the user's interaction outcome at step $j+1$ (or marker r_{j+1}), given this step's item. For the prediction of marker r_{j+1}, we first obtain the user's current relationship with the system items via Equation (24) to get a column vector $\mathbf{v}$. This vector is then concatenated with item-level features and passed through a fully-connected network with activation function ϕ_1 of choice so that both the item information and the user's relationship with the items can be encoded in Equation (25). The final prediction of marker $\hat{r}_{j+1}$ is computed via Equation (26) using another fully connected layer, where ϕ_2 can be an activation function of choice, such as sigmoid for binary markers.

$$\mathbf{v}_j = \left(\mathbf{a}_j^\top \mathbf{M}_j^v\right)^\top, \tag{24}$$

$$\mathbf{n}_{j+1} = \phi_1\left(\mathbf{w_r}^\top \left[\mathbf{v_j}; \mathbf{q^{id}}_{j+1}; \mathbf{q^f}_{j+1}\right]\right), \tag{25}$$

$$\hat{r}_{j+1} = \phi_2\left(\mathbf{w_p}^\top \mathbf{n}_{j+1} + \mathbf{b_p}\right). \tag{26}$$

5.5 Objective Function

The final loss $\mathcal{L}$ is the sum of the losses of activity time prediction $\mathcal{L}_t$, type prediction $\mathcal{L}_y$, and marker prediction $\mathcal{L}_r$, which are respectively defined as follows:

$$\mathcal{L} = \mathcal{L}_t + \mathcal{L}_y + \mathcal{L}_r, \tag{27}$$

$$\mathcal{L}_t = \sqrt{\frac{\sum_{i=1}^{N}\sum_{j=1}^{K_i}(\hat{t}_j^i - t_j^i)^2}{N \times \sum_{i=1}^{N} K^i}} \tag{28}$$

$$\mathcal{L}_y = -\sum_{i=1}^{N}\sum_{j=1}^{K_i}\sum_{k=1}^{|\mathcal{Y}|} w_k y_j^i \log \hat{y}_j^i \tag{29}$$

$$\mathcal{L}_r = -\sum_{i=1}^{N}\sum_{j=1}^{K_i}\left(w_c r_j^i \log p_j^i + w_r\left(1 - r_j^i\right)\right. \left.\log\left(1 - p_j^i\right)\right)\mathbb{1}_{r_j \neq -1}. \tag{30}$$

For time prediction, we use the mean squared loss $\mathcal{L}_t$. We use weighted cross entropy for type prediction loss $\mathcal{L}_y$. For the prediction of marker $\mathcal{L}_r$, we use binary cross-entropy [2]. For activities that have a missing marker r, we mask the loss via indicator function $\mathbb{1}_{r_j \neq -1}$. Similar to the performance prediction function defined in Equation (26), Equation (30) can be replaced by mean squared loss for numerical markers.

5.6 Computational Complexity of MoMENt

The computational complexity of MoMENt is an additive measure of the complexity of the RAU and MEMU components which are compatible with state-of-the-art RNNs, like LSTM and DKVMN. Specifically, for MEMU, for each sequence step, the time complexity is of $O(C^2 d_v + Cd_j + Cd_k + d_k d_j + Cd_h + d_k d_h)$, where C is the number of latent components (as in $\mathbf{M}^k$), d_j is the input e_{m_j}'s dimensionality, d_k is the number of latent variables for each component (as in $\mathbf{M}^k$), d_v is the user-item relationship features (as in $\mathbf{M}_j^v$), and d_h is RAU's hidden state size. For comparison, DKVMN has a computational complexity of $O(C^2 d_v + Cd_j + Cd_k + d_k d_j)$ for each step. Meaning that MEMU has an $O(Cd_h + d_k d_h)$ more computations compared to DKVMN. This additional computational need is because of the Equations (3), (4), and (6) in which the activity2marker influence is added to the model. Since C, d_k, and d_h are all latent features, they are usually set within a similar range. As a result, the additional term $O(Cd_h + d_k d_h)$ would be negligible compared to the overall computational complexity of DKVMN, especially its first term ($C^2 d_v$).

For the RAU component, the time complexity of each step is $O(d_h^2 + d_h d_j + Cd_h d_v)$. For comparison, other sequential models, such as LSTM or RNN have a complexity of $O(d_h^2 + d_h d_j)$ in each step. The term $Cd_h d_v$ is added to RAU's computational complexity because of having $\mathbf{M}_j^v$ in RAU to model the marker2activity influence. While this additional term is larger than LSTM's computational complexity, it is similar to that of DKVMN.

[2]Can be replaced by cross-entropy or regression loss for categorical or numerical r

Table 1. Descriptive Stats of the Datasets

Dataset	Activity Types	#. of users	#. of activities	#. of items	#. of tags	Missing Marker	Correct Answer	Avg (std). Inter-arrival Time
Junyi	Problem; Hint	2,063	658K	1,880	8	15.3%	68.8%	21.56 (57.54) mins
EdNet	Problem; Lecture;Explanation	709	282K	12,502	185	71.8%	42.5%	3.54 (29.5) mins
KDD-Algebra	Problem	574	809K	1,084	112	0%	74.5%	46.17 (85.13) mins
MovieLens	Rating	943	94K	1,682	19	0%	56.6%	5.79 (15.97) days
Amazon-Grocery	Rating	1,342	47K	7,342	126	0%	76.7%	55.40 (29.98) days
Yelp	Rating	1,988	106K	10,230	508	0%	66.0%	22.97 (17.45) days

Overall, the computational complexity of MoMENt would be $O(C^2 d_v + C d_h d_v + C d_j + C d_k + d_k d_j + d_k d_h + d_h^2)$ per step which is comparable to the state-of-the-art key-value memory-based recurrent models.

6 EXPERIMENTS

This section presents our experiments to evaluate the proposed model for user activity modeling [3]. We evaluate the model by assessing models' prediction performances on users' future activity timings, types, and markers (Section 6.4). An ablation study is also performed to explore the importance of our assumptions (Section 6.5). Finally, we present a qualitative analysis of MoMENt in capturing interpretable user-system relation that evolves over time, as well as user-system interaction dynamics temporally (Section 6.6). Before presenting the evaluation results, we first describe the datasets (Section 6.1), then the baseline approaches (Section 6.2), and the experimental setups (Section 6.3).

6.1 Datasets

We use the following six real-world datasets that contain users' online activities considering two tasks: (1) student learning and knowledge modeling in online courses and (2) item recommendation to online users. To ensure that TPPs have enough training trajectory, users with less than 20 activities are excluded from this study. The descriptive statistics of these datasets are provided in Table 1 and an overview of the datasets is given as follows.

Student Learning Datasets

- **Junyi Academy** [4] dataset comes from a Chinese e-learning website, which contains the trace data of students' learning, including attempting math problems and checking the hints. The system labels the learning material with math *areas*. These areas are used as tags in this dataset. In this and the following two student learning datasets, we use the outcome of attempting a problem (either correct or incorrect) as the marker.
- **EdNet** [5] is an AI tutoring platform that assists students in preparing for the TOEIC exam [6]. Four flavors of data sets with different detail levels are provided. We specifically use the KT4 set, which contains the most information, including student learning interactions with three types of learning materials: questions, lectures, and explanations.
- **KDD-Algebra** [7] is a dataset released for KDD Cup 2010 Educational Data Mining Challenge, which includes multi-type learning interactions of students with a computer-aided tutoring system for algebra. The labeled knowledge components are used as tags.

[3] Our code can be found at https://github.com/persai-lab/2024-TKDD-moment
[4] https://pslcdatashop.web.cmu.edu/Project?id=244
[5] https://github.com/riiid/ednet
[6] https://www.ets.org/toeic
[7] https://pslcdatashop.web.cmu.edu/KDDCup/

Recommendation Datasets

— **MovieLense-100k** [8] is a widely used benchmark dataset for recommender systems evaluations, which contains user ratings of movies and movie information, such as their genres. We use movie genres in this dataset as tags. The movie ratings ranging from 1 to 5 are used as markers. If a rating is greater than or equal to 3.5, we treat it as a positive marker; otherwise, negative. The same method is used for the following two datasets.
— **Amazon-Grocery** [9] is an e-commerce dataset that contains products in the department of Grocery and Gourmet Food from Amazon. Product categories are used as tags.
— **Yelp** [10] is another benchmark dataset for recommender system evaluations, which contains user reviews on food businesses, as well as business information such as location and categories, which are used as tags.

6.2 Baseline Approaches

In this section, we introduce 16 baseline approaches that are used in our evaluations. We consider six state-of-the-art temporal point process models. To better evaluate our proposed model under the aforementioned tasks, we also include another ten state-of-the-art recent developments in student modeling and recommendation systems.

Temporal Point Process Models. We consider the following state-of-the-art temporal point process models as baseline approaches, which will be evaluated on all six datasets.

— **Poisson Process** [36] is a classic TPP that models activities by assuming they happen at a constant rate.
— **GMHP** [72] is a multivariate Hawkes process model that represents each activity type as a dimension. The dependencies among types are captured by the Granger causality graph [14].
— **RMTPP** [17] is an RNN-based point process model that uses a standard RNN to model activity times and markers. It only models one set of markers, and the original framework uses activity type as the marker. Following this design, we use RMTPP–y and RMTPP–r to denote the model that respectively predicts learning material type and grade as markers.
— **ERPP** [71] is a similar approach that models activities and external features via two LSTMs. Similar to RMTPP, we consider two variations when the outcome (ERPP–r) and activity type (ERPP–y) are used as markers.
— **SAHP** [78] is one of the most recent neural-based Hawkes process models. It employs self-attention to capture the historical influences of activities on future ones. This model was originally designed for multi-type point processes and can be used to predict activity types (similar to treating type as a marker). For a more general marker prediction, we obtain the marker embedding in the same manner as the proposed event type embedding and consider the resulting variation SAHP–r as a baseline as well.
— **THP** [84] is a concurrent work to SAHP that employs a similar transformer architecture, within which a type-specific intensity function was proposed. Similarly, we also consider a variation THP–y by altering the intensity function as a marker-specific one, to obtain a more general marker prediction framework.

Student Learning Models. The following student activity learning and knowledge tracing models are used as baseline approaches and will be applied to the three student learning activity datasets for evaluation.

[8] https://grouplens.org/datasets/MovieLens-100k/
[9] http://jmcauley.ucsd.edu/data/amazon/
[10] https://www.yelp.com/dataset

— **DKT** [52] is the first attempt that integrates deep learning into student knowledge tracing while predicting student grades. Hidden states are learned as a summary of past knowledge via an RNN.
— **DKT-Forgetting** [47] is similar to DKT but uses time differences between exercises as extra features.
— **DKVMN** [76] is based on a Key-Value-Memory-Network that models static knowledge concepts and dynamic students' knowledge and performances over time.
— **HawkesKT** [61] is a recent first attempt at student knowledge tracing and performance prediction via point processes. Even though Hawkes intensity was adopted, the model is not designed to predict activity time nor considers multiple learning material types.
— **MVKM** [80] is the first student knowledge acquisition model that considers multi-type learning activities. Latent concepts are learned from tensor decomposition to represent the sequential orders of learning and are assumed to be shared across graded and non-graded activities.

Recommendation Models. We also consider the following Top-N recommendation models as baseline approaches for the evaluation of the model on the three recommendation datasets.

— **BPRMF** [53] is a matrix-factorization-based method that uses Bayesian Personalized Ranking objective function for the optimization.
— **GRU4Rec** [27] is a session-based sequential recommendation model that utilizes GRU to model the ranking scores.
— **SLRC** [62] is a collaborative filtering model that used the Hawkes intensity function to model the temporal aspects of historical activities.
— **TiSASRec** [41] is a sequential model that utilizes self-attention and inter-arrival times between historical activities.
— **KDA** [60] is a Fourier-based temporal method that is combined with knowledge graphs for item relation modeling.

6.3 Experimental Setup

TPPs typically use historical observations from a sequence to make meaningful and accurate future predictions of the same sequence based on the intensity function. We adopt this convention for modeling student learning activities, using the first 20 activities of all students as training, a randomly selected 20% of students and their later 20 activities as validation, and the remaining 80% of students and their later 20 activities as testing. For the task of recommendation, following the widely-adopted leave-one-out strategy in the literature [33, 41, 60, 63], we use the last positive item for testing, the second last item for validation, and the most recent 50 activities before validation as the training. Following the same convention, we randomly sample 100 negative items and generate recommendations with the ground truth item together based on the likelihood of each item being positive.

For the optimization, we use Adam [35] in the training process. Each model's hyperparameters are tuned separately using the validation set using Optuna [1]. Optuna is an open-source hyperparameter optimization framework that efficiently searches for the best hyperparameter values by parallelization and pruning. For batch size and initial learning rate, we search on respectively $\{16, 32, 64, 128\}$ and $\{0.01, 0.001, 0.0001, 0.00001\}$. For the proposed model MoMENt, the hidden state size for RAU is searched in $\{32, 64, 128, 256, 512\}$. Hidden state size for the fully connected network in $\{256, 512\}$, concept number C in $\{5, 10, 20, 50, 100\}$, d_v and d_k in $\{16, 32, 64, 128\}$. For both RMTPP and ERPP, the hidden state dimension is searched on $\{32, 64, 128, 256, 512, 1024\}$. In GMHP, the decay rate is searched in $\{1, 10, 50, 100, 500, 1000, 1500, 2000\}$. For each sequence, the best

decay rate that leads to the smallest negative likelihood is selected. For THP, the number of attention heads and the number of self-attention layers are respectively searched in $\{1, 2, 4, 8\}$, and d_v and d_k respectively in $\{16, 32, 64, 128\}$, the hidden state dimension is searched on $\{32, 64, 128, 256, 512\}$. For SAHP, the number of attention heads and the number of layers are respectively searched in $\{1, 2, 4, 8\}$, the hidden state dimension is searched on $\{32, 64, 128, 256, 512\}$, the hidden state dimension is searched on $\{32, 64, 128, 256, 512\}$. For DKT, we search the hidden state dimension in $\{32, 64, 128, 256, 512\}$. In EdNet, the best learning rate, batch size, and hidden state dimension are respectively 0.01, 32, 512, and in Junyi, they are respectively 0.01, 64, and 512, and in KDD they are respectively 0.01, 32, and 512. For DKVMN, We search memory size in the range of $\{1, 2, 5, 10, 20, 50\}$, state dimensions in $\{10, 50, 100, 200\}$, the hidden state dimension in $\{32, 64, 128, 256, 512\}$. For MVKM, we apply grid search on the student latent feature dimension K from 1 to 45 with step size 5, the question latent feature dimension C in $\{1, 2, \cdots, 9, 10\}$, the penalty weight ω in $\{0.01, 0.05, 0.1, 0.5, 1, 2, 3\}$, the Markovian step m in $\{1, 2, \cdots, 10\}$, and the learning resource importance parameter $\gamma^{[r]}$ in $\{0.05, 0.1, 0.2, 0.5, 1, 2\}$. in EdNet, the best student latent feature dimension, question latent feature dimension, penalty weight, and Markovian step are respectively 3, 3, 0.01, and 1. For BPRMF, GRU4Rec, and TiSASRec, we search the sizes of embedding and hidden state in $\{32, 64, 128, 256, 512\}$. For KDA, we searched the number of attention heads in $\{1, 2, 4, 8\}$, hidden state size in $\{32, 64, 128, 256, 512\}$, number of self-attention layers in $\{1, 2, 4, 8\}$ and attention hidden size in $\{1, 2, 4, 8\}$.

6.4 Model Evaluation

In this section, we present and evaluate model prediction performance. Given that all TPPs can be used to model point processes and thus can be adopted into different scenarios, they are evaluated in all six datasets. Additionally, all student learning models are evaluated using the three student learning datasets and all recommendation models are applied to the three recommendation datasets.

6.4.1 Performance Evaluation in Student Learning Datasets. In student learning datasets, the task of activity time prediction aims to predict the next study time of a student within the system, given their historical studying time and performance in the prior learning materials (up until step j). The goal of activity type prediction is to predict what kind of activity (e.g., a problem vs. a hint) a student will take at step $j + 1$, given the student's historical studying data. In marker prediction, the goal is to predict the student's performance (e.g., correct or incorrect) in the given learning material (item ID and features) at step j+1 and given their historical studying time and performance in the past.

To evaluate activity type and marker predictions, we use the **area under the receiver operating characteristic curve (AUC)** and **accuracy (ACC)**. For time prediction, we use **Root-mean-square deviation (RMSE)**. Since Algebra 2005-2006 only contains one activity type, no type prediction is presented for this dataset. Note that, except for the proposed model MoMENt, none of the baseline approaches can simultaneously predict activity time t, activity type y, and outcome r as the marker.

Model performance comparison is given in Table 2. First, we see that compared with all baselines, MoMENt achieves better student grade prediction (i.e., marker prediction) than all baseline approaches, showing the importance of modeling the influence of activity dynamics on marker distribution and vice versa. It can also be observed that variants of recent advanced TPPs such as SAHP–r and THP–r generally outperform state-of-the-art student models. This suggests that modeling the temporal aspects of student learning via intensity function improves the model's descriptive power of the data and therefore explains the superiority of these models.

Table 2. Performance Comparison in **Student Learning** Datasets

Dataset	Metric	MoMENt	SAHP	SAHP −r	THP	THP −r	RMTPP −y	RMTPP −r	ERPP −y	ERPP −r	Poisson	GMHP	HawkesKT	DKVMN	DKT	DKT - Forgetting	MVKM
Junyi	marker AUC	**0.837*****	-	0.765	-	0.758	-	0.680	-	0.715	–	–	0.730	0.785	0.742	0.735	0.769
	marker ACC	**0.855*****	-	0.771	-	0.764	-	0.742	-	0.760	–	–	0.754	0.810	0.725	0.742	0.776
	type AUC	**0.681****	0.677	–	0.640	–	0.500	–	0.510	–	–	0.668	–	–	–	–	–
	type ACC	**0.935****	0.901	–	0.875	–	0.780	–	0.846	–	–	0.716	–	–	–	–	–
	time RMSE	9.023*	10.716	9.771	11.204	11.109	14.645	14.371	10.150	8.900	54.378	8.851	–	–	–	–	–
EdNet	marker AUC	**0.898****	-	0.865	-	0.838	-	0.563	-	0.512	–	–	0.797	0.837	0.747	0.763	0.564
	marker ACC	**0.909*****	-	0.878	-	0.851	-	0.563	-	0.513	–	–	0.746	0.845	0.790	0.814	0.519
	type AUC	**0.704*****	0.505	–	0.674	–	0.563	–	0.604	–	–	0.425	–	–	–	–	–
	type ACC	**0.966*****	0.901	–	0873	–	0.853	–	0.898	–	–	0.517	–	–	–	–	–
	time RMSE	**7.890****	10.476	9.097	9.134	9.608	17.833	17.486	15.317	14.198	46.634	14.441	–	–	–	–	–
KDD-Algebra	marker AUC	**0.863*****	0.819		0.799		0.500		0.500		–	–	0.723	0.625	0.714	0.765	0.654
	marker ACC	**0.887*****	0.842		0.809		0.762		0.741		–	–	0.726	0.779	0.761	0.715	0.748
	time RMSE	**9.111***	9.744		10.342		11.19		11.08		73.781	38.781	–	–	–	–	–

Best (bold) and second-best (underlined) are highlighted. Statistically significant differences between MoMENt and the best on confidence levels of 99% (p-value < 0.01), 95% (p-value < 0.05), and 90% (p-value < 0.1) are respectively marked with ***, **, and *.

In terms of time prediction, compared with all TPPs that predict the next activity time, we observe that MoMENt achieves significantly lower time prediction RMSE in EdNet and KDD-algebra datasets. This shows that marker2activity modeling in MoMENt results in better activity time predictions compared to the other baselines. Only in Junyi GMHP achieves the smallest RMSE among all models. A possible reason is that in this dataset, student learning dynamics are less complex due to fewer concepts. Therefore, GMHP's simple but effective intensity function may capture the data dynamics well, resulting in more accurate time prediction. We should note that as another traditional TPP, Poisson process has the worst performance in time prediction, suggesting that activity arrivals follow a more complicated distribution than the "memoryless" constant rate. Another interesting observation is that TPP variants (e.g., SAHP−r) usually predict time better than their original model (e.g., SAHP), suggesting a strong association between marker r with time distribution and its importance in time prediction.

MoMENt is also shown to achieve better type prediction performance, usually by large margins. Combining with the model's performances in time prediction, it demonstrates that MoMENt can accurately capture activity arrival patterns, suggesting that successfully capturing activity2activity and marker2activity influences in RAU results in higher model capacity. To conclude, MoMENt is the first model that can simultaneously predict when (studying time), how (learning material type), and how much (grades) the students learn. It consistently outperforms all baseline approaches in terms of type and grade prediction across all datasets and all baseline approaches in terms of time prediction in more complex datasets. MoMENt's superiority suggests the effectiveness and importance of modeling the bidirectional influences between markers and activities.

6.4.2 Evaluating MoMENt on Recommendation Datasets. Table 3 shows the Top-N recommendation performance of all TPPs and recommendation baselines. Two widely used metrics namely **hit ratio (HR)** and **Normalized discounted cumulative gain (NDCG)** at 5 and 10 are used as the evaluation metrics for this experiment. As shown in the table, MoMENt outperforms all the baselines on all datasets in HR@5, HR@10, and NDCG@10. This shows the effectiveness of MoMENt in successfully capturing bidirectional influences to improve the recommendations. In the Yelp dataset, we see that KDA is the strongest baseline. A possible explanation is that this dataset has a more complex and larger tag system on all the items (i.e., restaurants) compared with the other two datasets. Since KDA uses customized knowledge graph embeddings as an additional source of knowledge, this complex tag system potentially helps KDA to have a more accurate representation of the items and outperform other baselines.

We also observe that among all baselines, the state-of-the-art temporal point processes such as SAHP−r and THP−r generally achieve more competitive performances compared to others. This

Table 3. Performance Comparison in **Recommendation** Datasets

Dataset	Metric	MoMENt	SAHP−r	THP−r	RMTPP−r	ERPP−r	TiSASRec	SLRC	KDA	BPRMF	GRU4Rec
MovieLens	HR@5	**0.428**	0.412	0.409	0.387	0.367	0.357	0.387	0.402	0.318	0.345
	HR@10	**0.609**	0.579	0.570	0.528	0.531	0.529	0.414	0.547	0.484	0.491
	NDCG@5	**0.266**	0.256	0.249	0.240	0.239	0.254	0.215	0.254	0.159	0.228
	NDCG@10	**0.317**	0.310	0.291	0.282	0.279	0.293	0.256	0.308	0.201	0.276
Amazon-Grocery	HR@5	**0.491**	0.479	0.481	0.471	0.418	0.386	0.395	0.433	0.410	0.419
	HR@10	**0.556**	0.530	0.538	0.529	0.519	0.481	0.452	0.533	0.501	0.505
	NDCG@5	**0.389**	0.332	0.364	0.354	0.323	0.281	0.252	0.267	0.269	0.289
	NDCG@10	**0.410**	0.389	0.401	0.396	0.382	0.321	0.294	0.398	0.333	0.316
Yelp	HR@5	**0.463**	0.401	0.392	0.408	0.413	0.284	0.402	0.422	0.380	0.280
	HR@10	**0.626**	0.567	0.554	0.549	0.537	0.535	0.609	0.617	0.598	0.460
	NDCG@5	0.299	0.267	0.250	0.267	0.259	0.278	0.264	**0.303**	0.258	0.175
	NDCG@10	**0.393**	0.345	0.317	0.324	0.348	0.375	0.378	0.381	0.377	0.233

Best (bold) and second-best (underlined) are highlighted.

Table 4. Ablation Study in Student Learning Datasets

Dataset	Metric	MoMENt	MoMENt\t	MoMENt\y	MoMENt-t\y
Junyi	marker AUC	**0.837**	0.826	0.831	0.816
	marker ACC	**0.855**	0.835	0.840	0.798
	type AUC	**0.631**	0.524	–	–
	type ACC	**0.935**	0.883	–	–
	time RMSE	**9.023**	–	9.904	–
EdNet	marker AUC	**0.898**	0.873	0.861	0.855
	marker ACC	**0.909**	0.875	0.863	0.856
	type AUC	**0.704**	0.677	–	–
	type ACC	**0.966**	0.959	–	–
	time RMSE	**5.890**	–	8.226	–
KDD-Algebra	marker AUC	**0.863**	0.849	–	–
	marker ACC	**0.887**	0.852	–	–
	time RMSE	**9.111**	–	–	–

Table 5. Ablation Study in Recommendation Datasets

Dataset	Metric	MoMENt	MoMENt\t
MovieLens	HR@5	**0.428**	0.398
	HR@10	**0.609**	0.577
	NDCG@5	**0.266**	0.236
	NDCG@10	**0.317**	0.272
Amazon-Grocery	HR@5	**0.491**	0.467
	HR@10	**0.556**	0.530
	NDCG@5	**0.389**	0.375
	NDCG@10	**0.410**	0.383
Yelp	HR@5	**0.463**	0.403
	HR@10	**0.626**	0.607
	NDCG@5	**0.299**	0.258
	NDCG@10	**0.393**	0.302

improved performance is possible due to an efficient historical activity2activity influence captured by self-attention weights while learning from the intensity function. Recent Top-N recommendation models that employ the intensity functions to capture time dependencies such as KDA and SLRC also achieve promising performances. We also see that more traditional approaches that either do not model time (GRU4Rec and BPRMF) or only capture the durations between activities while ignoring activity dynamics such as the decaying historical influence (TiSASRec) are only shown to achieve moderate performances across all recommendation datasets. These two observations together suggest the importance of modeling user temporal dynamics in accurately depicting the dynamics of user behavior.

In summary, MoMENt is the only method that models the temporal aspects of user activities via an intensity function that also captures important dependencies such as marker2activity and marker2marker influences, which possibly explain its better performance over baseline approaches across different datasets.

6.5 Ablation Study

In this ablation study, we assess the importance of *activity2marker* influence which has been largely neglected by the literature. We specifically consider MoMENt$\backslash t$, MoMENt$\backslash y$, and MoMENt$\backslash t\backslash y$ which are variants of the proposed model by respectively masking time input t, type input y, and both t and y from the full model MoMENt. Since the Algebra 2005-2006 dataset in the Knowledge Tracing domain and all datasets in the recommendation domain only have one type of activity, only the results of MoMENt$\backslash t$ are meaningful and presented.

The results of student activity modeling and top-N recommendation are shown in Tables 4 and 5, respectively. In the student learning datasets, the goal is to predict student future learning time,

type, and outcome (i.e., marker). Table 4 results show that MoMENt outperforms all variants across all datasets in all predictions. Comparing all the variants, we see that MoMENt$\backslash t \backslash y$ achieves the lowest performance of marker prediction. This result demonstrates that simultaneously modeling time, type, and marker are essential to capture the complex dynamics of student learning for all the prediction tasks. We also notice that in EdNet, masking type (MoMENt$\backslash y$) results in a less competitive marker prediction performance than masking time (i.e., MoMENt$\backslash t$), whereas, in Junyi, this observation is reversed. This is potentially related to the more types of learning materials that are provided in EdNet. Having more learning materials may lead to more complex knowledge dynamics in students, e.g., when students switch from one learning material type to another.

Next, we look at the task of Top-N recommendation results in Table 5. In this task, the models rank and return the top N items based on their likelihood of having a positive marker (i.e., a positive rating). We observe that the proposed MoMENt outperforms the variant MoMENt\t usually by large margins. Similar to student activity modeling, this result highlights the significance of activity time in representing user activity dynamics and marker distributions (positive or negative ratings). In other words, a user's future interactions with items are not only associated with the past items they interacted with but also with *when* they interacted with them.

To summarize, our ablation study demonstrates the importance of modeling activity dynamics in terms of time and type, as well as their relationship to marker distribution, which is consistent with our observations and conclusions in the model performance comparison in Section 6.4.

6.6 Interpretation and Analysis

In this section, we present different interpretations that MoMENt can provide with two case studies from the two application domains. Within each case study, the analysis is divided into two parts: we first demonstrate MoMENt's ability to capture a fine-grained interpretation of user-system relations over time. Furthermore, we present an analysis to show the temporal patterns of user interactions, where we analyze how users' historical behavior can have an influence on users' future activities.

6.6.1 Understanding Student Learning Dynamics in Online Courses. Modeling student learning activities while quantifying student knowledge is an essential task in the education domain. Understanding students' knowledge of course concepts can be beneficial for improving teaching or tutoring quality, where students' strengths and weaknesses can be better assessed. Although research has shown that learning is a multifaceted process, modeling and predicting students' studying time (or time to study) and learning material type, in relation to student knowledge and performance has not been fully studied in the literature. This section aims to demonstrate how to use the interpretation that MoMENt provides to fill this gap.

Consider an online course. Suppose there are C knowledge concepts covered by the course learning materials. Then, the features of those course concepts and the student's mastery of them at step j can be represented by the static key matrix $\mathbf{M}^k$ and dynamic value matrix $\mathbf{M}^v_j$, respectively. To obtain the student's knowledge representation in the concepts, we pass the learned value matrix $\mathbf{M}^v_j(z)$, namely the z^{th} row of the value matrix $\mathbf{M}^v$ at step j through Equation (24)–(26) using the learned parameters after training. The resulting value can be interpreted as the student's understanding of concept C_z at step j during practice. This step is iterated over all training steps $j \in [1, K]$ for all concepts $z \in [1, C]$ to obtain the student's changing knowledge over time.

A sample student from Junyi Academy and their evolving knowledge states of three latent concepts over 20 practicing steps are given in Figure 4. As it is shown in the figure, the student practiced five problems with Mean and Median tags and provided wrong answers in them (steps 1–5 marked in red (X) in the top row). During this practice, we notice that the student's knowledge

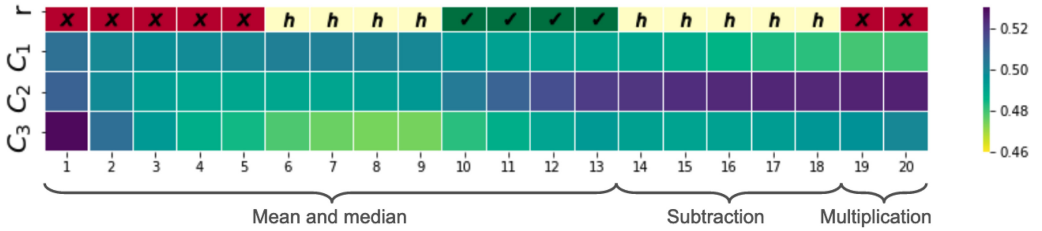

Fig. 4. Changing knowledge states of a sample student from Junyi over 20 practice steps (x-axis) w.r.t. concepts C_1, C_2, and C_3 (y-axis). The problem tags are given below the braces (e.g., multiplication problems for the last two steps). The markers r are given in the top row, with red (X), green (✓), and yellow (h) representing wrong answers, correct answers, and hint-checking, respectively. Cell shades in the three concept (bottom) rows represent the degrees of the student's understanding, ranging from yellow (low) to purple (high).

representation of concepts 2 and 3 (C_2 and C_3) starts to drop, which suggests that these two concepts are the most related to this question. A possible explanation is that by getting the wrong answers, the student got more and more confused over these concepts. Then, we observe that the student started to check the hints related to these tags (steps 6–9 in yellow (h)), probably wanting to learn how to answer the problems better. During this hint-checking process, their knowledge of these two concepts (especially concept 2) increases, which also suggests their relatedness to the problems. Furthermore, it indicates that checking these hints helps the student gain a better understanding of these concepts, to later obtain correct answers (steps 10–13 in green (✓)), and have a much faster knowledge gain. Later, when the student moves on to another tag (i.e., Subtraction), they only gain a marginally better understanding of concepts 2 and 3 over hints. This observation might suggest that these hints are not sufficient for this student's knowledge gain in those concepts. It may also explain why the student is not attempting to answer the Subtraction question at the end of the hint-checking, as they may not feel confident enough to do so, given their current knowledge. Another observation is that the student's understanding of concept 1 has been generally decreasing over time. One possible explanation is that not practicing the related questions to this concept frequently enough has led to the forgetting of this concept. This case study shows how one can visualize MoMENt's learned parameters to represent a student's knowledge gain over their practice activities, figure out the concepts they are struggling with, and find out the effective learning materials associated with different concepts for the student.

In addition to knowledge gain visualization, we also study the temporal aspects of student learning, which is studied and represented as the following two aspects: studying student learning intensities in continuous time, and temporal activity dependencies within the trajectory of student learning. Modeling the temporal aspects of student learning has been increasingly shown to be critical in capturing the true engagement of students in various important applications such as procrastination modeling [73] and the modeling of knowledge forgetting [58]. To examine student learning dynamics in continuous time, we present the sample student's learning intensities as a function of time while they are interacting with different learning materials (i.e., answering problems and checking hints). The activity sequence intensities of the sample student of Figure 4 is presented in the upper plot in Figure 5. Furthermore, to understand the temporal activity dependencies or the influence of students' historical learning on their future learning activities, we compute an influence matrix between activities. In this matrix, we show the contribution of each historical j^{th} activity on the later i^{th} activity where $i > j$. This lower-triangular influence matrix is presented in the lower plot in Figure 5.

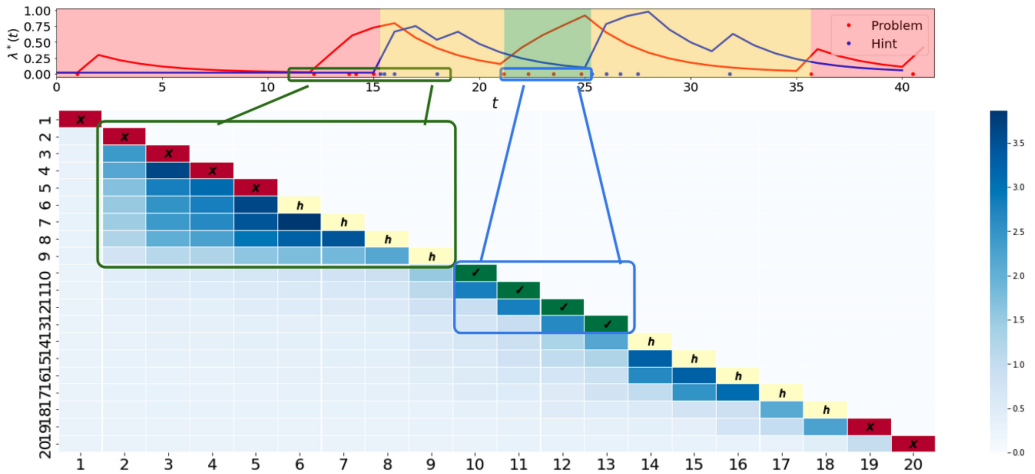

Fig. 5. Temporal patterns of student learning, represented by learning intensity (upper) and influence matrix of historical learning on the future (lower). Upper: intensities of student learning dynamics (y-axis) as a continuous function of time t (x-axis). Dots represent problem (red) or hint (blue) activity types happening in student sequence over time. The intensities of problem and hint activities (as a function of time) are shown by the continuous red and blue lines in the plot. The colored-shaded areas have the same meaning as the color-coded markers: red for the wrong problem attempts, green for the correct problem attempts, and yellow for hint checking. Lower: the influence matrix, showing influences of historical activities on the future ones for the same sample student during practice. The cell in j^{th} row and i^{th} column represents the influence of activity at step j on the later i^{th} activity, with a darker shape representing a higher influence. The markers of activities are provided on the diagonal, where red (X), green (✓), and yellow (h) respectively represent wrong answer, correct answer, and hint-checking.

More specifically, the upper plot in Figure 5 shows occurrences of problem (red) or hint (blue) activities of the sample student's sequence as dots on the X-axis. The X-axis represents continuous time, marking 0 as the time the student started their activities. The red (problems) and blue (hints) curved line plots show the modeled intensities of student learning dynamics ($\lambda^*(t)$ in Equation (16)) as continuous functions of time t. For example, when the student has intense problem-solving activities, the modeled problem activity intensity function (red curve) increases rapidly and drops gradually when the student switches to another activity type (e.g., hint-checking). Additionally, the plot area is color-coded according to the student performance markers. We use red for showing wrong problem attempts, green for correct problem attempts, and yellow for hint-checking activities.

For the influence matrix shown in the bottom part of Figure 5, we compute the expected intensity at the i^{th} activity that was contributed by the j^{th} activity [24] via the Hawkes component in Equation (16). Here, the influence of the j^{th} activity on the later i^{th} activity is represented by the cell at j^{th} column and i^{th} row in the influence matrix. The X and Y-Axes in this influence matrix show the time step of the student activity sequence, as opposed to the continuous time value presented in the upper plot of Figure 5. Additionally, we show the activity markers on the diagonal of the influence matrix, where red (X), green (✓), and yellow (h) respectively represent wrong answer, correct answer, and hint-checking. For example, the 6^{th} column of the influence matrix shows checking a hint at step 6 and its association with getting a correct answer in later steps.

As shown in Figure 5, the student's first few steps of practice have a high and lasting effect on the later activities. This phenomenon is represented by the darker cells under the diagonal of the

influence matrix in steps 2-9, shown within the green box. A possible explanation is that getting wrong answers in a row (steps 1-5) makes the student realize that they lack some knowledge in answering these problems (as also shown in steps 1-5 of Figure 4, where the student knowledge in C_2 and C_3 decreases). Therefore, the student starts to check the hints (steps 6-9) intensively, trying to understand the concepts related to their previous few attempts. This intensity is shown in the upper plot of Figure 5 by the steep increase in the hint-checking intensity curve (the blue curve) around time 15, which is aligned with attempt step 6 in the influence matrix. As the student continues practicing, the influences from the past wrong answers on the later activities start to decay. This is possibly because these previous problems start to have less and less associations with the student's later practices. This reduction in associations is shown by the fading shades on the columns of the influence matrix in the green box.

Another interesting pattern can be found in the student's later steps of learning, where the student answers multiple problems correctly (e.g., steps 10-13 in the blue box of the influence matrix). The upper plot indicates that the student has a steady pace during these steps, as shown by the unperturbed and moderately sloped problem activity intensity curve and the spaced-out red dots before time 25. In these attempts, the influence matrix shows that the learning activities have strong associations with their most recent prior activity and weaker associations with the earlier ones. A potential reason for this short-term influence could be the activity marker or the correctness of the student response. Namely, answering a question correctly has a small influence on the student's following few activities, probably because the student does not feel the urge to practice intensively or check hints, as opposed to when they needed to correct themselves after giving a wrong answer. Rather, the student is shown to try the next problems using a similar pace and get a good understanding of the problems (also suggested by the fast increase of knowledge in C_2 in Figure 4). The above analyses demonstrate how MoMENt can be used in providing interpretations of when and how students learn and the associations between their practice intensity, activity types, knowledge, and performance.

6.6.2 Understanding User Movie-Watching Dynamics. In the recommender system application, we also start with our analysis of the user-system relationship captured by MoMENt. For that, we analyze the user's evolving relationship to item latent components. We consider one sample user from the MovieLens dataset for visualization and analysis, as shown in Figure 6. Similar to Section 6.6.1, we use the z^{th} row of the value matrix $\mathbf{M}^v$ at step j and the learned parameters after training to obtain the user's evolving relation to each component C_z, which under this setting can be interpreted as the user's interest in different movie latent features. The ground truth markers are again provided in the top row of Figure 6, with red (X) and green ($\checkmark$) respectively representing negative and positive user ratings. The movie genres are represented as a heat map with a binary color scheme in the lower plot of Figure 6). Here, a bright cell at the j^{th} column and g^{th} row represents that the movie that the user watched at the j^{th} step has a genre of the g^{th} row, and a dark cell represents that it does not. For example, the last movie of this sample user is of the genres of Fantasy and Sci-Fi.

As shown in Figure 6, the user mostly tries Fantasy movies, especially at the beginning of their sequence. The user then tries a few Crime movies (steps 24-37) between fantasy movies. While the user is watching these Crime movies, their represented interest in component 3 (C_3) decreases, and once the user switches back to the fantasy movies it keeps staying low. This suggests that component 3 is likely to be mostly associated with some common elements in Crime movies (e.g., heist or robbery), and when this user is exposed to such elements repetitively, their general interest in component 3 decreases. This observation matches the ground truth markers where the user gives some negative ratings to the Crime movies during these steps. Unlike C_3, components 1 and

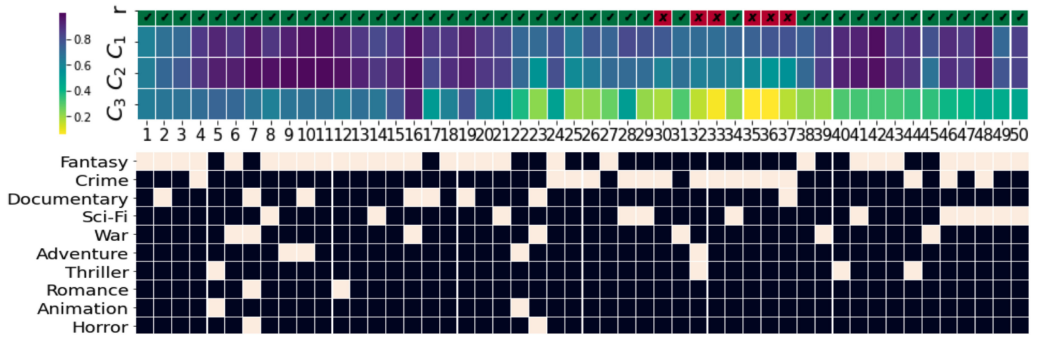

Fig. 6. Upper: The user's evolving interest w.r.t. three learned components of movies (y-axis) over 50 steps (x-axis). Markers, i.e., positive and negative ratings, are respectively shown in green and red on the top row of this plot. Lower: tags (genres) for the watched movies in the above 50 steps. A white cell in a row represents the movie of the genre of the corresponding row, and black represents the opposite. The top 10 genres are shown in this plot.

2 may be more related to the elements in Fantasy movies (e.g., magic and fairy tales) which matches their fast increase while the user is watching this genre and giving positive ratings to them.

To analyze the dynamics of user-system interactions, we present the influences of the historical movie-watching experience on the later ones (lower) and the intensity of movie watching (upper) of the sample user in Figure 7. The upper plot in Figure 7 shows occurrences of movie-watching activities of the sample user's sequence as dots on the continuous-time X-axis. The red curve plot shows the modeled movie-watching intensity function of the target user ($\lambda^*(t)$ in Equation (16)). The plot area is color-coded according to the user's feedback markers (green for positive and red for negative feedback). The influence matrix in the bottom part of Figure 7 is computed similarly to the knowledge tracing influence matrix. Similar to the color-coded areas of the upper plot, the activity markers are color-coded on the diagonal of the influence matrix, with red (X) for the negative and green ($\checkmark$) for the positive experience.

As shown in the figure, examples of two different influence patterns can be observed and presented in green and blue boxes. As we see in the green box of the influence matrix, the first several Fantasy movies that the user watches have a lasting and strong influence on the user's future choices of movies to watch. This is demonstrated by the darker blue cells in columns and rows 1 to 10 in the influence matrix. As an example of such influence, suppose that the user watches Harry Potter 1 which may consequently motivate them to watch the next few movies in the series to complete the story. During this time that the watching experience is positive, we also see a growing intensity trend, represented by the red intensity function curve in the upper plot. One explanation for this lasting effect and intense activities is that the user enjoyed the good movies from the past and, therefore is willing to try more within a shorter time (i.e., bursts of activity clusters). More interestingly, this phenomenon is also in line with the study that reveals the association between binge-watching-like behavior and the dopamine released by human brains [21], which possibly explains the high fast-growing intensity and interests (as shown in Figure 6) during these bursts of activities.

Another interesting pattern is shown in the blue box in Figure 7 where the influence of the past activities is not very strong and lasting, suggested by the darker area only close to the diagonal. In other words, these movies have less influence on the user's future choices of movies to watch. During the steps, we see that the user gives mostly negative ratings to the watched movies. Combined with the observation from the upper plot where the intensity of movie-watching activities

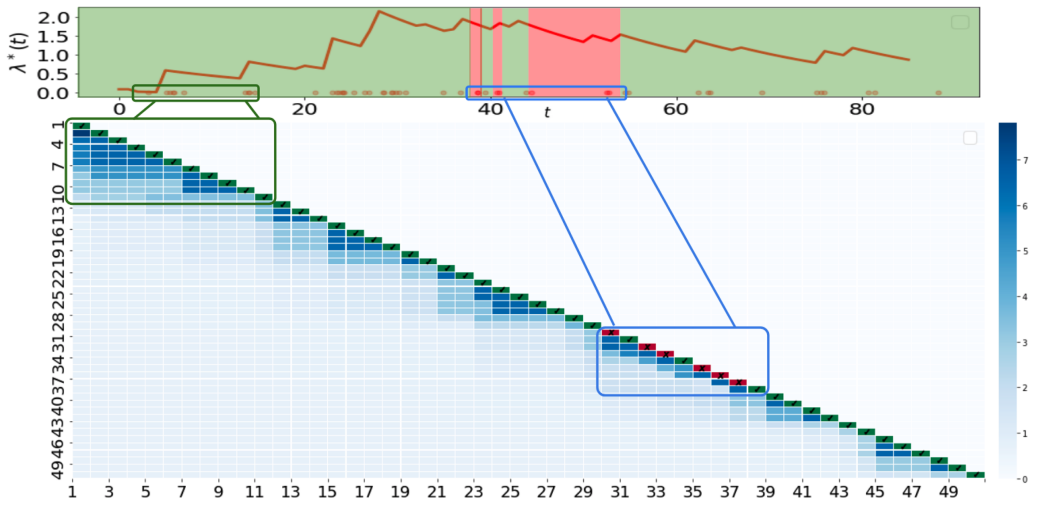

Fig. 7. Dynamics of user movie-watching experience. Upper: intensities of movie-watching (y-axis) as a continuous function of time (x-axis). Lower: the influence of historical movie-watching experience on the future. Cell in the j^{th} row and the i^{th} column represents the influence of the movie watched at step j on the later i^{th} movie, with a darker shade representing a higher influence. The markers are given on the diagonal, where green and red respectively represent positive and negative ratings.

decreased, a possible explanation is that after watching a Crime movie that the user did not enjoy, the user is less motivated to watch more movies. These discovered patterns show that MoMENt can be effective in providing meaningful explanations of when and how users interact with the system and the associations between their interaction dynamics, continuous time intensity, and interests.

7 DISCUSSION

In this section, we summarize the experiment results and discuss the limitations of MoMENt. The experiment results demonstrate that modeling the bidirectional influences of activity dynamics on marker distribution (activity2marker) and vice versa (marker2activity) is essential for better marker prediction. This was shown in Sections 6.4.1 and 6.4.2, where MoMENt outperformed the baseline methods in the tasks of student grade prediction and top-N recommendations, respectively. Similarly, our experiments in the education domain show the importance of these bidirectional influences in activity type predictions.

Likewise, the superior performance of MoMENt compared to the baselines in activity time predictions in the more complex datasets indicates the fundamental importance of bidirectional influence modeling. We noted that in one of the less complicated datasets (i.e., Junyi) one baseline (GMHP) achieves the smallest RMSE among all models. This result can be attributed to Junyi's more straightforward learning dynamics, with fewer concepts, and GMHP's simple but effective intensity function. However, we showed that overly simple and "memoryless" TPPs, such as the Poisson process, are inadequate to capture user activity dynamics. Additionally, GMHP does not have the ability to predict activity markers. Combining the results of MoMENt's performance in time, type, and marker predictions suggests that the RAU component can successfully capture the activity2activity and marker2activity influences, the MEMU component can effectively model activity2marker and marker2marker influences, and thus MoMENt can accurately capture activity arrival patterns with a higher model capacity.

We further studied the activity2marker influence in more detail via our ablation study by masking type and time inputs from the model. Our results in both application domains indicate the significance of activity time modeling for an accurate representation of user activity dynamics and marker distributions. Specifically, we showed that modeling time and type together is essential for better performance in all the prediction tasks. Similar to the results presented in Section 6.4.1, the importance of type vs. time modeling varies by the complexity of the datasets. For instance, the time component played a more important role in marker predictions in the less complicated dataset Junyi.

To obtain further understanding of what is learned by MoMENt, we performed an interpretation and analysis study in Section 6.6. Our visualization and analyses of the learned model demonstrate that MoMENt can be used to interpret when and how users interact with the system and the associations between their interaction dynamics, continuous time intensity, and underlying interaction factors. More specifically, visualization of MoMENt's learned components explains which items and activity types are associated with the discovered concepts and how a user's relationship with these components (and items) changes over time. Also, MoMENt's influence matrix illustrates which activities have long-lasting influences on other activities in accordance with activity feedbacks (markers). Coupled with the intensity plot, these visualizations can be used to demonstrate what types of activities and their feedbacks (markers) relate to intense vs. steady activity timings in users.

Together, these analyses can unveil remarkable observations. For example, our case study in the education domain showed that the target student's past failures had a large influence on their consequent fast-paced hint-checking activities to learn the missing concepts, followed up by the more steady and regularly-timed successful attempts on the learned concepts. In recommender systems, our analyses revealed how a user's positive experience in one genre led to more frequent subsequent consumption of the same genre, and their exploration of another genre with a negative experience was associated with less frequent and more spaced system interactions. These analyses showed that a user's future interactions with items are not only associated with the past items they interacted with but also with when they interacted with them.

Although MoMENt was shown to be successful in our experiments and interpretable in our analyses, there are some limitations in this work that suggest new directions for future work. For example, in the design of our input layer, we use one-hot encoding $\mathbf{e}_{\mathbf{r}_j}$ to represent markers, which can increase the memory complexity with a larger number of items and item tags, compared to using other embedding methods, such as hashing. Although in some of our baselines, e.g., in GRU4Rec [27], using one-hot encoding was shown to work better than using an embedding layer, exploring different embedding strategies in the input layer may show improvements in model performance or complexity.

Also, we designed the MEMU component to have a MANN-like structure and the RAU component to have an LSTM-like structure. Although these structures perfectly fit our applications and were shown to have superior performance compared to the baselines, our design is limited in not exploring other architectures for these two components. For example, while our experiments show that having a simpler LSTM-like structure in the MEMU component would not suffice to capture the detailed marker dynamics, we have not explored a more complex MANN-like structure for the RAU component. Exploring other designs for the MEMU and RAU components would be a worthwhile direction for future works.

Finally, as MoMENt relies on historical events to model the dynamics and perform predictions, it is not evaluated in the extreme settings, such as in the cold-start setting with shorter sequences. In the future, the model can be adapted, or novel models can be researched to perform well in such extreme settings.

8 CONCLUSIONS

In this paper, we proposed a novel marked point process model MoMENt that captures important fine-grained and interpretable bidirectional influences between activity dynamics and activity markers that have been overlooked by the literature. More specifically, our proposed Recurrent Activity Updater (RAU) component in MoMENt models complex user activity dynamics in terms of time and type while capturing the important activity2activity influence as well as the marker2activity influence. In parallel, MoMENt's Memory-Enhanced Marker Updater (MEMU) captures marker2marker and activity2marker influences. The key-value memory structure in MEMU provides a fine-grained explanation of the user interaction outcomes (marker) dynamics. Additionally, in the prediction layer of MoMENt, we proposed a flexible and interpretable activity time distribution model that can capture positive, negative, and neutral influences of past activities on future ones.

We formulated MoMENt according to the user activity modeling problem and applied it to the tasks of student activity modeling in online education systems as well as Top-N recommendations. MoMENt is the only model that has the ability to predict activity type, time, and marker at the same time. To evaluate MoMENt, we conducted extensive time, type, and marker prediction experiments comparing with 16 baselines as well as ablation studies on six real-world datasets and demonstrated its effectiveness in all the given tasks. Furthermore, to showcase the interpretability of MoMENt and the interrelationships it can capture between activity timings and markers, we presented two case studies in each of the student modeling and recommendation application domains. Our case study reveals interpretable representations of user-system relations over time and provides meaningful insights into how users' future interactions are influenced by their past experiences.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2623–2631.

[2] Massil Achab, Emmanuel Bacry, Stéphane Gaïffas, Iacopo Mastromatteo, and Jean-François Muzy. 2018. Uncovering causality from multivariate Hawkes integrated cumulants. *Journal of Machine Learning Research* 18, 192 (2018), 1–28.

[3] Preeti Bhargava, Thomas Phan, Jiayu Zhou, and Juhan Lee. 2015. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data. In *Proceedings of the 24th International Conference on World Wide Web*. 130–140.

[4] Veronika Bogina, Tsvi Kuflik, Dietmar Jannach, Maria Bielikova, Michal Kompan, and Christoph Trattner. 2023. Considering temporal aspects in recommender systems: A survey. *User Modeling and User-Adapted Interaction* 33, 1 (2023), 81–119.

[5] Qi Cao, Huawei Shen, Keting Cen, Wentao Ouyang, and Xueqi Cheng. 2017. DeepHawkes: Bridging the gap between prediction and understanding of information cascades. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1149–1158.

[6] Zi-Hao Cheng, Jian-Wei Liu, and Ze Cao. 2022. Hypergraph neural network Hawkes process. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–6.

[7] Sung Min Cho, Eunhyeok Park, and Sungjoo Yoo. 2020. MEANTIME: Mixture of attention mechanisms with multi-temporal embeddings for sequential recommendation. In *Fourteenth ACM Conference on Recommender Systems*. 515–520.

[8] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. 2019. DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Paris-Saclay Junior Conference on Data Science and Engineering (JDSE 2019)*.

[9] Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 3512–3520.

[10] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

[11] Daryl J. Daley and David Vere-Jones. 2007. *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Springer Science & Business Media.
[12] Daryl J. Daley and David Vere-Jones. 2007. *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Springer Science & Business Media.
[13] Saurabh Dash, Xueyuan She, and Saibal Mukhopadhyay. 2022. Learning point processes using recurrent graph network. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.
[14] Vanessa Didelez. 2008. Graphical models for marked point processes based on local independence. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70, 1 (2008), 245–264.
[15] Fangyu Ding, Junchi Yan, and Haiyang Wang. 2023. c-NTPP: Learning cluster-aware neural temporal point process. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. 7369–7377.
[16] Tim Donkers, Benedikt Loepp, and Jürgen Ziegler. 2017. Sequential user-based recurrent neural network recommendations. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 152–160.
[17] Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. 2016. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1555–1564.
[18] Nan Du, Yichen Wang, Niao He, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. *Advances in Neural Information Processing Systems* (2015), 3492–3500.
[19] Mehrdad Farajtabar, Yichen Wang, Manuel Gomez-Rodriguez, Shuang Li, Hongyuan Zha, and Song Le. 2017. COEVOLVE: A joint point process model for information. *Journal of Machine Learning Research* 18 (2017), 1–49. https://doi.org/10.5555/3122009.3122050
[20] Mehrdad Farajtabar, Jiachen Yang, Xiaojing Ye, Huan Xu, Rakshit Trivedi, Elias Khalil, Shuang Li, Le Song, and Hongyuan Zha. 2017. Fake news mitigation via point process based intervention. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 1097–1106.
[21] Maèva Flayelle, Natale Canale, Pierre Maurage, Claus Vögele, Laurent Karila, and Joël Billieux. 2018. Assessing binge-watching behaviors: Development of the "watching TV series motives" and the "binge-watching engagement" questionnaires. *Journal of Behavioral Addictions* 7 (2018), 70.
[22] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2330–2339.
[23] Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural Turing machines. *arXiv preprint arXiv:1410.5401* (2014).
[24] Alan G. Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.
[25] Alan G. Hawkes and David Oakes. 1974. A cluster process representation of a self-exciting process. *Journal of Applied Probability* 11, 3 (1974), 493–503.
[26] Bin He, Ting Luo, and Shan Huang. 2019. Product sustainability assessment for product life cycle. *Journal of Cleaner Production* 206 (2019), 238–250.
[27] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
[28] Chih-Hsiang Ho. 1991. Nonhomogeneous Poisson model for volcanic eruptions. *Mathematical Geology* 23, 2 (1991), 167–173.
[29] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[30] Roya Hosseini, Teemu Sirkiä, Julio Guerra, Peter Brusilovsky, and Lauri Malmi. 2016. Animated examples as practice content in a Java programming course. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*. 540–545.
[31] Rahma Innadia, Respatiwulan Respatiwulan, Siswanto Siswanto, and Yuliana Susanti. 2020. Estimation the number of road traffic accidents in Indonesia with a non-homogeneous compound Poisson process. In *AIP Conference Proceedings*, Vol. 2296. AIP Publishing.
[32] Martin Jacobsen. 2006. *Point Process Theory and Applications: Marked Point and Piecewise Deterministic Processes*. Springer Science & Business Media.
[33] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.
[34] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. 2010. Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the Fourth ACM Conference on Recommender Systems*. 79–86.
[35] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 13 pages. http://arxiv.org/abs/1412.6980

[36] J. F. C. Kingman. 2005. *Poisson Processes*. John Wiley & Sons, Ltd. https://doi.org/10.1002/0470011815.b2a07042. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470011815.b2a07042

[37] Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 447–456.

[38] Andrew S. Lan, Jonathan C. Spencer, Ziqi Chen, Christopher G. Brinton, and Mung Chiang. 2018. Personalized thread recommendation for MOOC discussion forums. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 725–740.

[39] Earl Lawrence, Scott Vander Wiel, Casey Law, Sarah Burke Spolaor, and Geoffrey C. Bower. 2017. The nonhomogeneous Poisson process for fast radio burst rates. *The Astronomical Journal* 154, 3 (2017), 117.

[40] 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*. 1419–1428.

[41] 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.

[42] Yang Li, Nan Du, and Samy Bengio. 2017. Time-dependent representation for neural event sequence prediction. *arXiv preprint arXiv:1708.00065* (2017).

[43] Scott W. Linderman and Ryan P. Adams. 2014. Linderman14. 32 (2014), 1–9. papers://d471b97a-e92c-44c2-8562-4efc271c8c1b/Paper/p659

[44] 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*. 6754–6764.

[45] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1400–1409.

[46] Swapnil Mishra, Marian-Andrei Rizoiu, and Lexing Xie. 2016. Feature driven and point process approaches for popularity prediction. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1069–1078.

[47] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The World Wide Web Conference*. 3101–3107.

[48] Amir Shareghi Najar, Antonija Mitrovic, and Bruce M. McLaren. 2014. Adaptive support versus alternating worked examples and tutored problems: Which leads to better learning?. In *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 171–182.

[49] Yosihiko Ogata and David Vere-Jones. 1984. Inference for earthquake models: A self-correcting model. *Stochastic Processes and their Applications* 17, 2 (1984), 337–347.

[50] Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. In *12th International Conference on Educational Data Mining, EDM 2019*. International Educational Data Mining Society, 384–389.

[51] Harold Pashler, Nicholas Cepeda, Robert V. Lindsey, Ed Vul, and Michael C. Mozer. 2009. Predicting the optimal spacing of study: A multiscale context model of memory. *Advances in Neural Information Processing Systems* 22 (2009), 1321–1329.

[52] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in Neural Information Processing Systems* 28 (2015), 505–513.

[53] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[54] R. Rotondi and E. Varini. 2019. Failure models driven by a self-correcting point process in earthquake occurrence modeling. *Stochastic Environmental Research and Risk Assessment* 33, 3 (2019), 709–724.

[55] Huawei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. 2014. Modeling and predicting popularity dynamics via reinforced poisson processes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 28.

[56] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. 2021. SAINT+: Integrating temporal features for EdNet correctness prediction. In *LAK21: 11th International Learning Analytics and Knowledge Conference*. 490–496.

[57] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. *Advances in Neural Information Processing Systems* 28 (2015).

[58] Behzad Tabibian, Utkarsh Upadhyay, Abir De, Ali Zarezade, Bernhard Schölkopf, and Manuel Gomez-Rodriguez. 2019. Enhancing human learning via spaced repetition optimization. *Proceedings of the National Academy of Sciences* 116, 10 (2019), 3988–3993.

[59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.

[60] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2020. Toward dynamic user intention: Temporal evolutionary effects of item relations in sequential recommendation. *ACM Transactions on Information Systems (TOIS)* 39, 2 (2020), 1–33.

[61] Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, and Shaoping Ma. 2021. Temporal cross-effects in knowledge tracing. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 517–525.

[62] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling item-specific temporal dynamics of repeat consumption for recommender systems. In *The World Wide Web Conference*. 1977–1987.

[63] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: Knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 109–118.

[64] Chunpai Wang, Siqian Zhao, and Shaghayegh Sahebi. 2021. Learning from non-assessed resources: Deep multi-type knowledge tracing. *International Educational Data Mining Society* (2021), 195–205.

[65] Weiqing Wang, Hongzhi Yin, Xingzhong Du, Quoc Viet Hung Nguyen, and Xiaofang Zhou. 2018. TPM: A temporal personalized model for spatial item recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)* 9, 6 (2018), 1–25.

[66] Yifan Wang, Yifang Qin, Fang Sun, Bo Zhang, Xuyang Hou, Ke Hu, Jia Cheng, Jun Lei, and Ming Zhang. 2022. DisenCTR: Dynamic graph-based disentangled representation for click-through rate prediction. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2314–2318.

[67] Xilin Wen, Jianfang Wang, Xu Yang, and Qiuling Zhang. 2021. A sequential recommendation of Hawkes process fused with attention mechanism. In *2021 IEEE 23rd Int. Conf. on High Performance Computing & Communications; 7th Int. Conf. on Data Science & Systems; 19th Int. Conf. on Smart City; 7th Int. Conf. on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 301–308.

[68] 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.

[69] Jibang Wu, Renqin Cai, and Hongning Wang. 2020. Déjà vu: A contextualized temporal attention mechanism for sequential recommendation. In *Proceedings of The Web Conference 2020*. 2199–2209.

[70] Shuai Xiao, Junchi Yan, Mehrdad Farajtabar, Le Song, Xiaokang Yang, and Hongyuan Zha. 2017. Joint modeling of event sequence and time series with attentional twin recurrent neural networks. *arXiv preprint arXiv:1703.08524* (2017).

[71] Shuai Xiao, Junchi Yan, Xiaokang Yang, Hongyuan Zha, and Stephen M. Chu. 2017. Modeling the intensity function of point process via recurrent neural networks. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (San Francisco, California, USA) *(AAAI'17)*. AAAI Press, 1597–1603.

[72] Hongteng Xu, Mehrdad Farajtabar, and Hongyuan Zha. 2016. Learning Granger causality for Hawkes processes. In *International Conference on Machine Learning*. PMLR, 1717–1726.

[73] Mengfan Yao, Shaghayegh Sahebi, and Reza Feyzi Behnagh. 2020. Analyzing student procrastination in MOOCs: A multivariate Hawkes approach. *International Educational Data Mining Society* (2020).

[74] Mengfan Yao, Siqian Zhao, Shaghayegh Sahebi, and Reza Feyzi Behnagh. 2021. Relaxed clustered Hawkes process for procrastination modeling in MOOCs. *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)* (2021), 4599–4607.

[75] 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.

[76] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th International Conference on World Wide Web*. 765–774.

[77] Lu-Ning Zhang, Jian-Wei Liu, Zhi-Yan Song, and Xin Zuo. 2022. Temporal attention augmented transformer Hawkes process. *Neural Computing and Applications* (2022), 1–15.

[78] Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. 2020. Self-attentive Hawkes process. In *International Conference on Machine Learning*. PMLR, 11183–11193.

[79] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. 2015. Seismic: A self-exciting point process model for predicting tweet popularity. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1513–1522.

[80] Siqian Zhao, Chunpai Wang, and Shaghayegh Sahebi. 2020. Modeling knowledge acquisition from multiple learning resource types. In *Proceedings of The 13th International Conference on Educational Data Mining (EDM 2020)*. 313–324.

[81] Wang-Tao Zhou, Zhao Kang, Ling Tian, and Yi Su. 2023. Intensity-free convolutional temporal point process: Incorporating local and global event contexts. *Information Sciences* (2023), 119318.

[82] Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. 2020. Sequential modeling of hierarchical user intention and preference for next-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 807–815.

[83] 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.

[84] Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. 2020. Transformer Hawkes process. In *International Conference on Machine Learning*. PMLR, 11692–11702.

Received 15 January 2023; revised 30 December 2023; accepted 9 February 2024