High-capacity and interpretable temporal point process models for user activity sequence modeling

Mengfan Yao

University at Albany, State University of New York, miley.yao@gmail.com

The University at Albany community has made this article openly available. Please share how this access benefits you.

Follow this and additional works at: https://scholarsarchive.library.albany.edu/legacy-etd

Part of the Computer Sciences Commons

Recommended Citation


This Dissertation is brought to you for free and open access by the The Graduate School at Scholars Archive. It has been accepted for inclusion in Legacy Theses & Dissertations (2009 - 2024) by an authorized administrator of Scholars Archive.

Please see Terms of Use. For more information, please contact scholarsarchive@albany.edu.
HIGH-CAPACITY AND INTERPRETABLE TEMPORAL POINT PROCESS MODELS FOR USER ACTIVITY SEQUENCE MODELING

by

Mengfan Yao

A Dissertation
Submitted to the University at Albany, State University of New York
in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Philosophy

College of Engineering and Applied Sciences
Department of Computer Science
Summer 2022
To my beloved family and friends
ABSTRACT

A temporal point process can be viewed as a collection of random points falling in the space of time, which is a special type of stochastic processes that is used to model complex event sequences in continuous time. As event data has become more widely available, temporal point process models (TPPs), i.e. techniques for modeling temporal point processes, have been used to solve a wide range of real-world problems, in domains such as e-commerce, online education, and social media. Motivated by the limitations in TPP literature, this dissertation aims to explore and study the following research questions: 1) Focusing on the limitation of modeling temporal point processes independent and identically distributed (i.i.d), we want to study the question of how to model the relationship among temporal point processes. 2) To address the problem of ineffective representation of time dependencies, the second research question we want to answer is how to effectively represent external stimuli in temporal point process modeling. 3) To tackle the limitation and the scarcity of temporal point process modeling with markers, we want to answer the research question of how to efficiently model markers in temporal point processes while capturing the interrelationship between markers, activity timings and types. 4) Motivated by the difficulty of causality modeling in continuous time, we want to explore how to identify Granger causality in temporal point processes with complex data dynamics. Answering these questions will improve not just the field of temporal point process modeling in terms of model capacity and interpretation, but also potentially a wide range of disciplines with substantial societal impacts, such as in finance, health, recommendation, and education. To answer these research questions, we develop a series of novel TPPs with high capacity in event prediction as well as meaningful interpretations of data dynamics for a variety of real-world problems.
ACKNOWLEDGMENT

I’d like to express my heartfelt gratitude to my Ph.D. advisor and committee chair, Dr. Shaghayegh (Sherry) Sahebi, for providing me an opportunity to work under her guidance, for her endless support, encourage, and even more importantly, for always believing in me especially when I was lost in finding my directions in both research and life.

I am lucky to have my incredible committee, Dr. Mei-Hwa F. Chen, Dr. Petko Bogdanov, and Dr. Ming-Ching Chang who have provided me all-round valuable advice and suggestions throughout my Ph.D. candidacy. I am grateful to Dr. Mei-Hwa for her excellent suggestions, which have helped me clarify certain ambiguities in my dissertation and highlight the value of my research. I’d like to express my gratitude to Dr. Petko Bogdanov for his insightful questions, which have significantly helped in the improvement of my dissertation and presentation. I’d also like to express my gratitude to Dr. Ming-Ching Chang for all of his encouragement, support, and suggestions for improvement since the beginning.

In addition, I’d like to convey my gratefulness to Dr. Charalampos Chelmis, who served as my first advisor throughout the first two years of my Ph.D. and showed and taught me all a young Ph.D. student could hope to learn. I am grateful for his guidance, support, and patience.

I’d like to thank my dearest friends inside and outside the department, Zhao Siqian, Wang Chunpai, Mark Seabury, Angeliki Kapodistria, Dr. Yasitha Liyanage, Dr. Ehab Albadawy, and Dr. Mehrdad Mirzaei for their unwavering support, countless help, and the kindest company throughout this journey.

I’d want to express my gratitude to my parents, Yao Liwei and Sun Yanjun, whose memories of my youth are more vivid than mine and whose descriptions of my early years were invaluable. I would not be where I am now today if it weren’t for their unconditional love and support.

Finally, to John, Gurley and Cookie. I thank you for your patience, support, and love, for bringing me joy and happiness, for accepting me completely as who I am, and for helping and witnessing me becoming a better version of myself, slowly but surely.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>x</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>xii</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Limitations of TPPs</td>
<td>3</td>
</tr>
<tr>
<td>1.1.1 The I.I.D. Assumption in TPPs</td>
<td>3</td>
</tr>
<tr>
<td>1.1.2 Ineffective Representation of Time Dependencies in TPPs</td>
<td>4</td>
</tr>
<tr>
<td>1.1.3 Inefficient Modeling of Activity Markers in TPPs</td>
<td>5</td>
</tr>
<tr>
<td>1.1.4 Difficulty of Granger Causality Modeling in TPPs</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Research Questions</td>
<td>6</td>
</tr>
<tr>
<td>1.3 Contributions</td>
<td>6</td>
</tr>
<tr>
<td>1.4 Outline</td>
<td>8</td>
</tr>
<tr>
<td>2. Related Work</td>
<td>9</td>
</tr>
<tr>
<td>2.1 Overview of Traditional Point Process Models</td>
<td>9</td>
</tr>
<tr>
<td>2.2 Overview of Neural Point Process Models</td>
<td>10</td>
</tr>
<tr>
<td>2.3 Capturing Relationship Among Temporal Point Processes</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Modeling Complex External Effects in Temporal Point Processes</td>
<td>12</td>
</tr>
<tr>
<td>2.5 Marked Temporal Point Process Models</td>
<td>12</td>
</tr>
<tr>
<td>2.6 Granger Causality Modeling in Temporal Point Processes</td>
<td>13</td>
</tr>
<tr>
<td>2.7 Applications of TPPs</td>
<td>14</td>
</tr>
</tbody>
</table>
3. Modeling Cluster Structure in Hawkes Processes ........................................... 18
   3.1 Introduction ................................................................................................. 18
   3.2 Proposed Method ......................................................................................... 19
      3.2.1 Uni-Variate Hawkes Intensity Function .................................................. 19
      3.2.2 Relaxed Clustered Hawkes ..................................................................... 20
      3.2.3 A mixture Gamma prior ......................................................................... 21
      3.2.4 Objective Function .................................................................................. 21
      3.2.5 Optimization ........................................................................................... 22
   3.3 Experimental Evaluation .............................................................................. 27
      3.3.1 Datasets .................................................................................................. 27
      3.3.2 Experiment Setup .................................................................................... 29
      3.3.3 Baseline Approaches .............................................................................. 29
      3.3.4 Experimental Evaluation I: Estimated Parameters on Simulated Data ...... 30
      3.3.5 Experimental Evaluation II: Clustering Structure of Hawkes Parameters 31
      3.3.6 Experimental Evaluation III: Returning Time Prediction ....................... 32
   3.4 Understanding User Activity using RCHawkes-Gamma ............................... 34
      3.4.1 Studying Student Procrastination Behavior ............................................. 35
      3.4.2 Studying User Checking-in Behavior ...................................................... 37
   3.5 Conclusion ................................................................................................... 40

4. Modeling External Stimuli in Hawkes Processes ............................................. 42
   4.1 Introduction .................................................................................................. 42
   4.2 Proposed Method .......................................................................................... 43
      4.2.1 Parameterization of External Stimuli and Self-excitement ...................... 43
      4.2.2 Objective Function .................................................................................. 47
      4.2.3 Parameter Inference ................................................................................ 49
   4.3 Experimental Evaluation .............................................................................. 49

vi
4.3.1 Datasets ......................................................... 50
4.3.2 Baseline Approaches .............................................. 51
4.3.3 Experiment Setup ..................................................... 51
4.3.4 Experimental Evaluation I: Parameter estimation ..................... 52
4.3.5 Experimental Evaluation II: Time Prediction ............................. 53
4.3.6 Experimental Evaluation III: Ablation Study on the Importance Ex-
ternal Stimuli .............................................................. 56
4.4 Discovering Procrastination Patterns via SSHP ....................... 58
4.4.1 Clusters of Learning Dynamics Captured by SSHP ...................... 58
4.4.2 Association between Procrastination Behaviors and Grades ............. 61
4.5 Applying SSHP to Recommendation Datasets .......................... 62
4.6 Conclusion ...................................................................... 63
5. Modeling Marked Temporal Point Processes ............................ 65
5.1 Introduction .................................................................... 65
5.2 Problem Formulation ....................................................... 67
5.3 MoMENt ...................................................................... 69
  5.3.1 Input Layer ............................................................. 70
  5.3.2 Memory-Enhanced Marker Updater (MEMU) ......................... 71
  5.3.3 Recurrent Activity Updater (RAU) ..................................... 73
  5.3.4 Prediction Layer ......................................................... 75
    5.3.4.1 Activity Time Prediction ............................................ 75
    5.3.4.2 Activity type prediction ............................................. 76
    5.3.4.3 Marker Prediction .................................................... 76
  5.3.5 Objective Function ...................................................... 77
5.4 Experiments .................................................................... 78
  5.4.1 Datasets .................................................................. 78
  5.4.2 Baseline Approaches .................................................... 79
6.5 Conclusion ................................................................. 116
7. Conclusion ................................................................. 117
  7.1 Limitations ............................................................ 119
  7.2 Future Work .......................................................... 120
# LIST OF TABLES

1.1 A summary of models developed in this dissertation .......................... 8

3.1 A summary of baseline approaches. .................................................. 30

3.2 RMSE (±error standard deviation) of \( \hat{A} \) and \( \hat{U} \) on partially missing test set and completely missing test set, with various missing data ratios (\( r \)) .......................... 31

3.3 Spearman’s correlation between learned parameters and computed normalized student delays. \( p<0.001^{***} \) \( p<0.01^{**} \) \( p<0.05^* \) .......................... 36

3.4 Kruskal Wallis test on delays in different clusters in CANVAS dataset. \( p<0.001^{***} \) \( p<0.01^{**} \) \( p<0.05^* \) .......................... 36

3.5 Kruskal Wallis test on delays in different clusters in MORF dataset. \( p<0.001^{***} \) \( p<0.01^{**} \) \( p<0.05^* \) .......................... 36

3.6 Clusters of venue categories identified in Foursquare-NYC dataset. ........ 38

3.7 Clusters of venue categories identified in Foursquare-TKY dataset. ........ 38

4.1 A summary of baseline approach and the proposed SSHP ....................... 51

4.2 RMSE of parameters learned by proposed model SSHP in synthetic datasets. . 52

5.1 descriptive stats of the datasets ..................................................... 79

5.2 Performance comparison in Student Learning datasets. Best (bold) and second-best (underlined) are highlighted. Statistically significant differences between MoMENt and the best on confidence levels of 99% and 95% are respectively marked with *** and **. .......................... 84

5.3 Performance comparison in recommendation datasets. Best (bold) and second-best (underlined) are highlighted. .......................... 85

5.4 Ablation study in Knowledge Tracing datasets. .................................. 86

5.5 Ablation study in recommendation datasets. ..................................... 86
6.1 Evaluating model ability to identify Granger causality in terms of AUC, Kendall’s \( \tau \) coefficient and Spearman coefficient in simulated datasets. . . . . . . . . . . . 107

6.2 Time prediction comparision of all models in terms of RMSE. . . . . . . . . . . . . . . . . . 113
LIST OF FIGURES

3.1 The ground truth of $A$’s correlation matrix (a), and the estimated $\hat{A}$’s correlation matrix learned by each model. ......................................................... 32

3.2 Time prediction error and 95% confidence interval on synthetic partially missing test set (left) and completely missing test set (right) datasets with varying data missing ratios ($r$) ................................................................. 33

3.3 Time prediction error in partially missing test set (left) and completely missing test set (right) data with 95% confidence interval respectively on student learning datasets (upper) and user check-in (lower) datasets. ............................... 33

3.4 Box plot of user check-in times in Foursquare-NYC. ........................................ 39

3.5 Box plot of user check-in times in Foursquare-TKY. ........................................ 40

4.1 Activity intensity of a sampled student-assignment pair. ................................. 50

4.2 Predicted Intensity of a synthetic sequence. .................................................. 53

4.3 Time prediction RMSE on partially missing test set with 95% confidence interval on Syn-10 and Syn-90. ................................................................. 54

4.4 Time prediction RMSE on completely missing test set with 95% confidence interval on Syn-10 and Syn-90. ................................................................. 54

4.5 Time prediction RMSE on partially missing test set with 95% confidence interval on two real-world datasets. ................................................................. 55

4.6 Time prediction RMSE on completely missing test set with 95% confidence interval on two real-world datasets. ................................................................. 56

4.7 Time prediction RMSE with 95% confidence interval of SSHP and variations on partially missing test set. ................................................................. 57

4.8 Time prediction RMSE with 95% confidence interval of SSHP and variations on completely missing test set. ................................................................. 57

4.9 Clusters of student learning dynamics characterized by SSHP in CANVAS. .... 59
4.10 Clusters of student learning dynamics characterized by SSHP in MORF.

4.11 Clusters of student learning dynamics characterized by SSHP in CANVAS and MORF.

5.1 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.

5.2 An illustration of MEMU.

5.3 An illustration of RAU.

5.4 Changing knowledge states of a sample student from Junyi dataset over 20 steps of practice (x-axis) w.r.t concepts $C_1$, $C_2$, and $C_3$ (y-axis). The tags of the problems over 20 steps are given below the parentheses (e.g. multiplication problems for the last 2 steps). The markers $r$ are given in the top row, with red, green, and yellow respectively representing wrong answers, correct answers, and hint-checking. The shades of cells in the three bottom rows represent the degrees of student’s understanding, ranging from yellow (low) to purple (high).

5.5 Temporal patterns of student learning represented by learning intensity (upper) and influence of historical learning on the future (lower). Upper: intensities of student learning dynamics (y-axis) as a continuous function of time $t$ (x-axis), and the colored-shaded areas have the same meaning as the color-coded markers. Lower: influences of historical activities on the future ones for the same sample student during practice. Cell in $j$-th row $i$-th column represents the influence of activity at step $j$’s influence 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, green, and yellow respectively represent wrong answer, correct answer and hint-checking.

5.6 Upper: evolving user’s interest w.r.t 3 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 of movies where a white cell in a row represents the movie is of the corresponding row of genre while black represents the opposite. Top 10 genres were shown in this plot.
5.7 Dynamics of user moving-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 $j$-th row $i$-th column represents the influence of 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.

6.1 Granger causality evaluation in the simulated self-exciting dataset: Visualizations of the ground truth (a), the predicted Granger causality recovered by the proposed model REACT (b) and baseline approaches (c-f). The cell in $d$-th row and $d'$-th column being white and black respectively represents that dimension $d'$ Granger cause and does not Granger cause dimension $d$.

6.2 Granger causality evaluation in the simulated self-inhibiting dataset: Visualizations of the ground truth (a), the predicted Granger causality recovered by the proposed model REACT (b) and baseline approaches (c-f). The cell in $d$-th row and $d'$-th column being white and black respectively represents that dimension $d'$ Granger cause and does not Granger cause dimension $d$.

6.3 Granger causality in Junyi estimated by REACT.

6.4 Granger causality in IPTV estimated by REACT.

6.5 Ablation study result of REACT.
CHAPTER 1

Introduction

The rapid growth of modern technologies and online systems such as social media, online merchandise, and e-learning generates a vast volume of user-system interaction data on a daily basis. A user activity sequence, or a sequence of interactive activities collected from a user, contains not only rich information about these interactions, such as activity timestamps, activity type, activity markers (i.e., related auxiliary features), but also hints about user activity patterns such as their social relationship [40, 41, 133], purchase preferences [26, 34, 55, 61, 103, 119, 121], and learning patterns [71, 80, 86, 135, 136]. As a result, powerful models that can effectively characterize these user activity sequences and discover user behavioral information can be beneficial for a wide range of domains and applications. However, user activity sequences are characterized by activity timestamps that are randomly spread over continuous time. Modeling such activity sequences could be challenging because it requires models that can effectively describe data dynamics in continuous time based on only discrete observations.

Temporal point processes, on the other hand, provide feasible solutions to this challenge. A temporal point process, in a nutshell, can be viewed as the set of points that fall randomly in continuous time and are ideal for representing a collection of activities that are characterized by their arrival times (i.e., random points that fall in the space of time). Temporal point process models (TPPs) are methods that describe and model these point processes (e.g., user activity sequences) via conditional density functions. A conditional density function, is a function of time $t$ conditioning on history of the process happened before time $t$, denoted as $f(t|\mathcal{H}_t)^1$ with $\mathcal{H}_t$ representing this history. Then the joint density function for the realization of historical observations $\{x^1, ..., x^K\}$, where $x^\tau$ is the time when $\tau$-th activity arrives, can be obtained as follows:

$$f^\ast(x^1, ..., x^K) = \prod_{\tau=1}^{K} f(x^\tau|\mathcal{H}_{\tau-1}) \quad (1.1)$$

1Often denoted as $f^\ast(t)$ for simplification
The above function is one way to characterize a particular point process, however could be difficult for model design and interpretability [27]. Alternatively, conditional intensity function $\lambda(t|H_t)$ representing the intensity distribution of activity arrivals based on the history $H_t$ have been more commonly adopted, which is shown to be a function of $f(t|H_t)$ and its corresponding cumulative distribution function $F(t|H_t)$:

$$\lambda(t|H_t) = \frac{f(t|H_t)}{1 - F(t|H_t)} = \frac{f(t|H_t)}{1 - \int_{x_n}^t f(s|H_{x_n})ds}$$  \hspace{1cm} (1.2)

Based on the formulation of the conditional intensity function $\lambda(t)$, two categories of modern TPPs have been widely proposed and applied in the machine learning community, i.e. traditional and neural point processes respectively.

In one category, there are traditional TPPs that model the arrivals of activities via a kernel-specified intensity function (e.g. an exponential function), that encodes one’s belief of how the dynamics of activities change over time [49, 57]. Traditional TPPs excel at data interpretation thanks to their interpretable parameterizations that can be customized for different assumptions. However, such customization may be a double-edged sword: misspecification can reduce model predictive power in fitting future activities. On the other hand, with the rapid advancement in deep learning, the last decade has witnessed a trend of neural point processes which model point processes via Recurrent Neural Networks (RNNs) [32, 82, 104, 127, 129, 149]. Neural TPPs are better suited for learning complex dynamics and unknown distributions. However, inheriting the limitations of methods such as neural networks, neural TPPs usually require more training data, and are less interpretable than traditional TPPs.

The advantages and disadvantages described above for traditional and neural TPPs, that is, the trade-off between clear model interpretations and flexible model specifications, is aligned with the central problem of machine learning namely the trade-off between model interpretability and model capacity (e.g. predictive power). As evidenced by recent TPPs, the improvements made in traditional and neural TPPs have attempted to leverage models’ advantages (e.g. by replacing RNN with LSTM to further improve models’ predictive power) [82, 129]. Or, more recently, to alleviate such trade-off by improving traditional TPPs’

\footnote{Often denoted as $\lambda^*(t)$ for simplification}
predictive powers (e.g. with the introduces of new intensity functions), or by improving neural TPPs’ interpretations in explaining the data (e.g. by using more interpretable framework such as self-attention) [142, 149].

To this end, this dissertation aims to develop a series of temporal point process algorithms with high model capacity and that can be interpreted, as possible solutions to several challenging problems that have not been fully studied in the TPP literature.

1.1 Limitations of TPPs

In this section, we would like to first explore several gaps and limitations in TPP literature.

1.1.1 The I.I.D. Assumption in TPPs

A common assumption that has been used in TPPs is that the point processes at hand are identical and independently distributed (i.i.d). As a result, it is common practice to model each sequence independently, with no consideration given to possible differences among them and their relationships. When discussing the dependency in one event sequence, different types of activities within the sequence (e.g. a user’s browsing activities vs purchasing activities) are sometimes assumed to have influence to each other (e.g. triggering or inhibiting). This type of relationship is usually captured and modeled by multi-dimensional point process models, in which each type of activities is modeled as a separate dimension and the dependencies between dimensions can be described by intensity functions concerning all dimensions [126, 130, 134, 146]. One could argue that this framework can be extended and applied to the modeling of multiple user activity sequences, to capture the relationship among user activity sequences. However, in many real-world applications such as recommendation or students learning behavior modeling, it is not always reasonable to assume a user’s actions to have direct influences on the behaviors of another random user. On the other hand, capturing relationships such as similarities or group structure of users will be beneficial for applications such as personalized recommendation and unseen data prediction [70], which is usually unrealistic for general TPPs with the i.i.d. assumption when the history is unobserved. To summarize, the literature of TPPs lacks powerful models that can effectively
model multiple processes and possible relationships among them.

1.1.2 Ineffective Representation of Time Dependencies in TPPs

It is essential to model time dependency in temporal point processes, i.e., the interrelationship between activities at different time. In many cases, time dependency is described by two types of effects: internal effects, which describe the time dependencies between future and historical activities within a point process, and external effects, which describes the dependencies between external environment and activities. Not only this decomposition has been shown to fit many real-world applications [34, 40, 41, 119, 135], it has also been shown to be effective in providing useful data interpretation. For example, in education domain, learning external effects can help instructors better understanding students reactions to the deadlines (i.e., a type of external effect exert on students), as a reflection of their time management skills, which in turn will be useful in understanding and detecting students’ procrastination behaviors [135]. For another example, capturing the internal effects can be useful to identify Youtube videos that potentially will trigger more follow-up clicks in the future [100], which could be beneficial for marketing purposes such as content promotion. Both types of effects are typically represented as hidden states or functions in neural TPPs, which are difficult to interpret and visualize by design. In traditional TPPs on the other hand, the representations of external effects in many TPPs are still either too simplistic (e.g., parameterizing an external trigger as a constant) or do not take into account the characteristics of the applications at hand. For example, students’ activities could be triggered by deadlines, but purchasing behavior might be related to others reasons such as the change of seasons. Modeling these two scenarios as the same is not ideal given the different natures of these domains. In fact, in the modeling of user activity sequences, understanding how users behave as a result of the characteristics of the external environment could be important in delivering high user satisfactions, especially for applications such as recommendations, or online course design. To summarize, TPPs that can be used to effectively represent both internal and external effects are still sparse.
1.1.3 Inefficient Modeling of Activity Markers in TPPs

In many real-world applications, it is common to have temporal auxiliary features associated with user activities over time. In the context of point processes, such additional information is usually referred to as markers. A common choice of marker that has been widely adopted in the literature is the activity type. However, depending on the applications, markers can also be other features such as the ratings of the purchases provided by the users or student grades in online courses for example. Often in TPPs, markers are not the focus and are discarded in the modeling based on the assumption that there is no relationship between marker and activity dynamics. This could potentially reduce model predictive power and interpretability, especially when the distributions of markers and activity timings are complex and are not independent from each other. Marked point processes models, on the other hand, are used for the modeling of aforementioned scenarios, where each activity is characterized by both of its timestamp and associated markers. An approach used in marked point process models is to empirically parameterize the intensity function as a function of the markers based on empirical observation of the distribution of markers and activity timestamps. However, this approach suffers the following limitations: 1. Such parametric representation usually requires domain knowledge, and therefore can be a challenge for the design of sufficient and accurate parameterizations. 2. It describes the distribution of timestamps as a function of markers, which only captures one-directional relationship such as how markers affect the arriving dynamics of activities, but ignores the possibility that marker distributions may also vary according to the distributions of activity timestamps. The modeling of this bi-directional interrelationship, on the other hand, not only can be strongly tied to model’s predictive power, but also can provide additional information that helps explain the data. In summary, the literature lacks TPPs that can efficiently model markers and the bi-directional influences between markers and activity arrival dynamics.

1.1.4 Difficulty of Granger Causality Modeling in TPPs

Identifying and learning Granger causality (GC) in point processes is an important but challenging task. On one hand, in many real-world applications, we are facing a large amount of user activities that are of different types, and modeling GC in user event sequences can go beyond merely the prediction and forecasting of future activities, but providing useful
interpretation such as the temporal patterns of the data. On the other hand, GC has mainly been used in time series with democratization [12, 21, 48, 65, 77, 111]. However, unlike time series, TPPs concerns activities modeling in continuous time, which makes the characterization of GC more difficult. Consequently, properly defining Granger causality in the context of temporal point processes remains a challenging task. Partially owing to these challenges, the literature of GC modeling for temporal point processes is extremely sparse. The full focus has been given to the learning of GC in multi-dimensional Hawkes process [1,37,130], while GC modeling in point processes with a broader type of dynamics is still unexplored.

1.2 Research Questions

Motivated by the aforementioned limitations and challenges, we want to investigate the following research questions, and explore their possible solutions throughout this dissertation.

- **RQ.1**: Focusing on the limitation of modeling temporal point processes i.i.d., we want to study the question of how to model the relationship among temporal point processes.

- **RQ.2**: To address the problem of ineffective representation of time dependencies, the second research question we want to answer is how to effectively represent external stimuli in TPPs.

- **RQ.3**: To tackle the limitation and the scarcity of temporal point process modeling with markers, we want to answer the research question of how to efficiently model markers in TPPs while capturing the interrelationship between markers, activity timings and types.

- **RQ.4**: Motivated by the difficulty of causality modeling in continuous time, we want to explore how to identify Granger causality in temporal point processes with complex data dynamics.

1.3 Contributions

To address the aforementioned limitations, we aim to develop interpretable and high-capacity TPPs that takes into account the following aspects:
Effective modeling. Focusing on the aforementioned limitations, we aim to develop a series of novel TPPs with high capacity in point process prediction. In particular, we propose a novel framework in Ch. 3 that effectively represents the relationship among Hawkes processes which is not restricted by the i.i.d assumption. Via the proposed group structure module, our model is able to predict future data dynamic even when the history is not observed. Taking the prior knowledge into consideration, we impose a Gamma prior that is shown to improve the learning performance. In Ch. 4, motivated by the oversimplified parameterizations in Hawkes processes, we provide a novel stimuli-sensitive Hawkes process model that is shown to more accurately recover the underlying data dynamic. Utilizing the similar structure in Ch. 3, this model is also able to predict unseen sequences’ returning times with low errors. Considering more complicated data dynamics, in Ch. 5 and Ch. 6, we respectively propose two novel neural point process models that can flexibly represent different types of data dynamics via LSTMs. More specifically, in Ch. 5, focusing on the limitation that markers are often discarded as irrelevant information, we propose a marked point process model that effectively represents marker distributions and their bi-directional relationship with event timings and types in continuous time. In Ch. 6, we propose a neural Granger causality point process model without relying on the strict linear additive formulation that has been strictly used in the literature, therefore can be flexibly extended to any general neural point process framework. Utilizing the group lasso regularizer, the model is shown to effectively recover Granger causality within and among point processes but less sensitive to noises.

Meaningful interpretation. An important goal we want to achieve in this dissertation is to propose effective temporal point process models that can also provide meaningful interpretations. With this goal in mind, in Ch. 3, we design a relaxed cluster module that is shown to effectively recover group structure of the data that provides extra interpretations, e.g., different user profiles identified based on their interacting patterns. In Ch. 4, we model the external stimuli exert on the processes and use them to interpret how the events react in accordance to the external environment, which is shown to bring meaningful interpretations in student procrastination analysis. In Ch. 5, focusing on the limitation of modern neural point process models that are hard to interpret, we propose a Memory-Enhanced Marker Updater module that can provide fine-grained interpretation of the bi-directional relationship between markers and event timings and types, which is also shown to be useful in applications such as analyzing user preferences and tracing student knowledge. The model proposed
in Ch. 6 is designed to recover the dependencies in point processes among dimensions from the viewpoint of causality in order to answer important questions such as why the events are unfolding in time in certain ways.

**Broadening and benefiting the domain applications.** Considering the fitness of the models to these applications, their importance and their potential societal impacts, as well as the limitations of state-of-the-arts designed for these applications, we consider the following applications: 1) student procrastination analysis in online courses (Ch. 3-4); 2) student learning dynamic modeling and knowledge tracing in online tutoring systems (Ch. 5-6); 3) recommendations considering the modeling and analysis of user digital foot prints logged by their digital devices (Ch. 3-4), user ratings and satisfactions in e-commerce platforms (Ch. 5), and user TV-watching behaviors (Ch. 6). These applications are related to a versatile challenging but important real-world problems, and tackling these problems goes a long way in making positive impacts to the society such as the enhancement of learning and teaching quality, the improvement of user satisfaction and wellness, which demonstrates the importance and the usefulness of this work.

### 1.4 Outline

In the rest of this dissertation, we will first conduct a literature review in Ch. 2 regarding traditional TPPs (Ch. 2.1) and neural TPPs (Sec. 2.2), focusing on the aforementioned limitations (Sec. 2.3-2.6), and with some important applications of TPPs (Sec. 2.7). The proposed solutions to our research questions are given in Ch. 3-6, summarized in Tabel 1.1. Finally, a conclusion is given in Ch. 7.

<table>
<thead>
<tr>
<th>RQ</th>
<th>Ch.</th>
<th>Task</th>
<th>Model</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>Model cluster structure in Hawkes processes</td>
<td>Relaxed Clustered Hawkes process (RCHawkes-Gamma)</td>
<td>Student procrastination analysis; User check-in behavior analysis</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Model external stimuli in Hawkes processes</td>
<td>Stimuli-Sensitive Hawkes Process (SSHP)</td>
<td>Student procrastination analysis</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>Model markers and relationship to activity dynamic in temporal point processes</td>
<td>Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt)</td>
<td>Knowledge tracing and analysis; User ratings modeling and analysis</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>Model Granger causality in temporal point processes</td>
<td>Recurrent Granger Causality in point process (REACT)</td>
<td>Student learning behavior analysis; TV-watching behavior analysis</td>
</tr>
</tbody>
</table>

**Table 1.1:** A summary of models developed in this dissertation
CHAPTER 2
Related Work

2.1 Overview of Traditional Point Process Models

As stated in the previous section, a point process can be characterized by the intensity function, often denoted as \( \lambda^*(t) \). Traditionally, an intensity function with customized parameterization is usually used to capture the underlying dynamics of the sequences. Based on the intensity functions, some popular point processes have been proposed, including Poisson processes [67], Hawkes processes (or self-exciting processes) [49], self-correcting processes [57], with more details described below.

**Poisson Processes** are the most classic point processes, 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 \( \mu \), 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 functions customized to applications [53, 56, 72, 105].

**Hawkes Processes** or self-exciting processes are the most popular and commonly used family of point processes. In a Hawkes process, activities are assumed to be “self-exciting”, meaning that the historical activities have a triggering effect on the future ones, characterized by intensity function \( \lambda(t) = \mu + \sum_{t_r < t} \phi(t - t_r) \), where \( \mu \) is the base rate which describes the activities naturally arrive as a result of external factors, and \( \phi \) 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 [50], which divides activities into immigrants and offspring, with the former representing the activities that are externally triggered by the environment (i.e. \( \mu \)), and the latter representing the activities that are self-excited within the process, (characterized by \( \phi \)). Thanks to their flexible and widely applicable formulation, as well as their clear interpretation, Hawkes process models with different intensity functions have been proposed and applied to many practical machine learning tasks based on different kernels functions.
\( \phi \), such as exponential (e.g. [34]), Gaussian (e.g. [130]), power-law (e.g. [84]), and their variations (e.g. [122]).

**Self-inhibiting Processes**, unlike Hawkes processes, assume that the historical activities have an inhibiting or regulating effect on future ones, characterized by intensity function 
\[ \lambda(t) = \exp(\mu t - \sum_{t_r < t} \alpha) \]
where \( \mu \) and \( \alpha \) are both 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-inhibiting itself from being too irregular. With this assumption, self-correcting process models are most popularly used to describe well-dispersed activity occurrences, for example, the modeling of earthquakes after aftershocks [90, 101].

### 2.2 Overview of Neural Point Process Models

Traditional TPPs with clear intensity function specifications excel at interpretations as one can analyze the dynamic of the sequences based on the learned parameters that have clear meanings. However, for the same reason, they are sometimes challenged by model misspecification. As a solution to this challenge, techniques such as recurrent neural networks and variants have been proposed to assist the modeling of point processes.

Particularly, to obtain a more flexible formulation of the intensity function of point processes, hidden representations of the neural networks such as RNNs are used to describe the time dependencies (e.g. self-exciting), and are used to formulate the intensity function usually via a non-linear function to capture complex dynamics [32, 82, 127]. As a result, because the parameterizations are not tailored to a specific application, neural TPPs and their formulations are relatively easier to be adopted from one domain to another. Despite their flexibility, the lack of interpretation is intrinsically tied to this ease of formulation. To tackle this limitation, attention mechanism has been integrated with neural point process models. For example, in [127], Xiao et al. proposed to add attention to RNN as the modeling of multi-dimensional point processes where the attention weight is used to describe the mutual excitation among related types of activities. Similar attempts have been made in [123], where a coverage is introduced to reduce possible misallocation of attention. Motivated by transformers [115], more recently self-attention framework and multi-head attention frameworks have also been introduced in neural RNN to characterize intensity function [142, 149].
2.3 Capturing Relationship Among Temporal Point Processes

In most TPPs it is common practice to assume that there is no relationship share among multiple sequences. As a result, sequences are usually modeled as independent and identically distributed (i.i.d), and the intensity function of a sequence is learned based only on its own history. However, the i.i.d assumption is not only often violated in real-world applications such as for user activity sequence modeling, it also requires a sufficient amount of activity observations to have meaningful and accurate predictions especially for users who have little or no activities. This is not ideal or even inhibited in practical problems such as timely user-product recommendation or students’ low-performance intervention. To tackle this limitation, some TPPs have been proposed to model the relationship among users based on the assumption that user behaviors can be categorized into a few prototypical types. These methods attempt to capture the similarities shared among users by establishing the connection between user interactions and collaborative filtering-based models. For example, Du et al. proposed to formulate users and purchases of online products as a matrix consists of user-product pairs, i.e. the collection of activities that are associate with a user and a product, which naturally leads to the matrices of parameters in the intensity function, where each cell in the matrix represents the parameterization of the corresponding user-product pair [34]. By assuming user behaviors can only be categorized into a limited number of prototypical types, they pose a low-rank constraint on each parameter matrix. Similar methods with different parameterizations or different low-rank structures have also been proposed in [55, 103]. Despite the success of these methods, the cluster structure of users is not directly modeled. Taking a completely different route, some feature-based clustering point process models have been proposed, combining point processes with Bayesian non-parametric priors (BNPs), for example, Dirichlet process [102]. However, the focus of these methods is either to cluster activities instead of user sequences [33, 81, 110], or to cluster user sequences relying on their auxiliary features [78]. To the best of our knowledge, modeling point processes with the clustering method in the case of user event sequences has rarely been explored.
2.4 Modeling Complex External Effects in Temporal Point Processes

In traditional TPPs, external effects on the arrival patterns of activities are usually modeled via base rate functions that in many cases are parameterized as constant. To capture more complex dynamics, base rate functions are usually customized based on the preliminary analysis of the activity distributions, or based on the prior knowledge of the applications. For example, in [10, 11], Bao et al. modeled blogs being forwarded as Hawkes processes, where the base rate is defined as the summation of blog’s decaying social influence, and the periodical rise-and-fall nature of user logging in to the platform, which is respectively parameterized by an exponential function and a shifted sine function. In another example, Ding et al. proposed to use power-law distribution to describe the reducing effect of videos to users watching behaviors over time [30]. Similarly, in [100], Rizoiu et al. proposed to use the number of video shares scaled by a constant coefficient $\mu$ to be the base rate of the Hawkes process with the goal to model the intensity of YouTube videos being watched. As another example, reinforced Poisson process [105] has been proposed to model the dynamics of a paper’s citations over time, where the rate function is defined as a log-normal function w.r.t time that captures the aging effect in attracting new attentions [113], scaled by a scalar that represents the attractiveness of the paper, and the attention the paper gets over time. In neural TPPs on the other hand, external stimuli and their effects are usually represented as biases that are shared over time, which works similar to representing base rate as a constant over time. Nevertheless, in many traditional and neural TPPs, what are the triggers in the external environment and how they affect the dynamics of user activities are not extensively studied.

2.5 Marked Temporal Point Process Models

Marked point processes are a special type of point processes where each activity is characterized by both its timestamp and associated features (i.e. markers). In the literature, marked point process models usually directly use markers to parameterize the activity intensity function, assuming that activity intensity is affected by the marker distributions. For example, Mishra et al. assume that a Twitter user’s social influence decides the popularity of their tweets, thus parameterized the intensity of a user’s tweets being retweeted as
a function of the user’s number of followers as a proxy of their influence (i.e. markers) [84].

A similar approach was proposed by Zhuo et al, where the modeling of the number of followers and timestamps are realized by Cox process [144]. This method of parameterization typically necessitates domain knowledge and preliminary research, because the properties of markers and how they affect activity intensities may vary with the applications. More recently, several neural marked point process models have been proposed which ease this requirement. For example, Du et al. proposed to model point processes via a RNN, where the hidden state at each time step is a non-linear function of both marker embedding and activity embedding, which captures the effect of both historical markers and activity timestamps [32]. In later attempts [129], Xiao et al. extended this framework by using a LSTM to model activity timestamps and markers using another independent LSTM. In the literature of neural marked point process models, a common approach is to sum up the representation of markers and activity timestamps at each time step for the prediction of the next activity. However, the markers and activity timestamps usually do not directly participate in the modeling of each other. Or in other words, the interrelationship between markers and activity timestamps and how their historical dynamics affect each other’s future is not directly modeled. Furthermore, as another limitation, all the above approaches model markers via standard RNNs, which forcefully represent the historical information as an abstract vector, which is hard to interpret.

### 2.6 Granger Causality Modeling in Temporal Point Processes

The literature of Granger Causality modeling in point processes is very sparse, partly due to the difficulty of extending its definition to the case of continuous time from its definition in time series [25]. In the limited number of related prior studies, Granger causality has mostly been modeled in multivariate Hawkes processes, with the assumption that activities are self-exciting. Focusing on Hawkes processes, Eichler et al. first connect the definition of Granger Causality with the structure of memory kernel functions [37]. Within this framework, a Granger Causality graph was built and used to define the intensity function to describe the triggering effects among types of activities. Later, Xu et al. extended this framework by adding a sparse group lasso regularizer to decrease model’s sensitivity to the noise in the data [130]. More recently, Cai et al. proposed a framework that aims to cap-
ture Granger causal relations among both dimensions as well as the sequences themselves, targeting the specific scenario where sequences are related such that signals can propagate among them and influence each other [16]. However, for the self-exciting assumption, these models are limited in their abilities to handle other types of time dependencies such as self-inhibiting or more complicated data dynamics. As another limitation, Granger causality in multivariate point processes has only been defined under the framework of traditional Hawkes process [37]. Several base functions are usually linearly added together to characterize the intensity function [16, 130, 143]. As a result, prior work has primarily used the this linear formulation even in neural point process models, to ensure that the identified between-dimensional relation is properly defined under the definition of Granger causality provided by Eichler et al [37]. For example, Zhang et al. recently proposed the first Granger causality neural point process model CAUSE [143]. An intensity function modeled by linearly additive base functions is adopted from traditional Hawkes processes [130], so that Granger causality could still be defined under this framework. For the aforementioned reason, this framework can not be easily extended to more general neural point processes especially when different choices of intensity functions are desired.

2.7 Applications of TPPs

Temporal point processes have been successfully used in a wide range of domains. Popular domains include information diffusion modeling (e.g. [40, 41, 133]). Other popular applications range from popularity prediction (e.g. [4, 84, 144]), financial applications (e.g. [8, 76]), disease evolution modeling (e.g. [3, 39, 99, 127]), and even violence or crime detection (e.g. [36, 85]).

More recently, thanks to the rich trace data that has been made available by the rising online tutoring systems, temporal point process have gradually been receiving more attention in the domain of education for tasks such as the modeling of students learning behaviors [15, 71, 86], the learning of knowledge representation [80, 108, 112], or procrastination modeling [135–138]. A similar trend has also been witnessed in the domain of recommendation, with applications including purchase time prediction (e.g. [26, 34, 55, 61, 103, 119, 121]), and personalized item recommendation (e.g. [34, 119, 120]). Despite TPPs’ great potential for the modeling tasks in the education and recommendation domains with the following
important and challenging problems under-explored.

**Student Procrastination Modeling.** As the online learning resources becoming prevalent, a growing number of students has started seeking education via online courses or tutoring systems [96]. However, many online users suffer from procrastination and issues with self-regularized learning, which are shown to be associated with negative educational and well-being side effects. Therefore, it is beneficial to have a comprehensive understanding of procrastination for the enhancement of education quality. Modeling student activities in continuous time and predicting their next study time are important problems that can help in creating personalized timely interventions to reinforce this goal. Past research has mainly described student procrastination by summarizing student activities into static features [18,94], which cannot fully represent the dynamics of students’ behavior through time. These studies aim to evaluate the relationships between these time-related features with student performance and do not model temporal aspects of procrastination [2,9,18,64]. For example, Asarta et al. examined students’ log data from an online course and use measures such as anti-cramming, pacing, completeness, etc to describe procrastination [5]. However, such methods are static and can not describe students’ varying behaviors over time. For another example, Park et al. classify students into procrastinators and non-procrastinators by formulating a measure via a mixture model of per-day student activity counts during each week of the course [92]. However, this is not able to model non-homogeneously spaced deadlines in a course. As none of these models consider the timing of students’ activities, they are not able to predict when the future activities will happen. Sequential data modeling via point process could potentially deal with this limitation, however, it has not been extensively studied and applied to the challenge of student procrastination modeling.

**Student Knowledge Tracing,** as another important problem in the education domain [93], aims to quantify students’ knowledge in course concepts as they interact with different learning materials. It is an essential problem in online courses, as it could provide instructors useful information of students learning behaviors and qualities, such as their understanding of the materials, and how it is changing over practicing over time. Understanding this problem can help instructors help students developing effective and efficient learning [45,91,95,141], which goes a long way in the improvement of teaching and tutoring qualities. Despite the success of these recent advances in knowledge tracing models (KTs), there are still important issues that haven’t been fully discussed in the literature. One under-studied area is modeling
the reciprocal influence of time and student knowledge. In most KTs, student activities are only modeled sequentially over time based on their orders. However, the actual timestamps which indicate when the students studied are generally discarded. As a result, two activities that have the same order from two sequences are treated as the same temporally, even though they may have a very different time distance to the previous activity. This difference can be used to model common situations such as the degrees of forgetting. Despite a few recent efforts to address this issue [22, 45, 106], this problem has not been comprehensively addressed. Another gap in the current KT literature is ignoring the effect of non-graded learning activities, such as watching lecture videos or checking explanations of the quiz answers. Most KTs only consider the graded materials (such as questions) as they provide observed feedback that can be used for knowledge estimation. On the other hand, however, TPPs have great potentials to model multi-type student learning activities for the problem of student knowledge tracing, thanks to their capacity to model activity time and type in continuous time. That being said, the efforts that have been made using TPP to tackle the aforementioned limitations of KT literature are still extremely sparse.

**Recommendation.** Early works of recommendation systems have mainly focused on the predictions of user preference of items (e.g. integer ratings), via collaborative-filtering-based methods [13, 51, 63, 69, 97]. However, the primary goal of such models is usually the optimization for user static one-time ratings, without considering the fact that users’ preferences or needs can be changing over time. Consequently, the temporal patterns of user-item historical interactions, which carry important clues of user dynamic preferences such as their satisfactions are overlooked by these models. Sequential recommendation models, on the hand, provide a solution to this limitation, by utilizing sequential models such as Markov Chains (MC) [98], or more recently RNN and variants [31, 74, 124, 139, 147]. Despite the improvement of such models over traditional CF-based recommendations in capturing temporal aspects, the literature of sequential recommendations that models the distributions of users time-varying preferences in continuous time, characterized by the actual timestamps of user activities like TPPs, is very sparse [20, 125, 148]. However, the time aspect could be of imperative importance in generating a timely effective recommendation. For example, for a user who usually watches movies on weekends but prefers to read on weekday nights, recommending a book to the user on weekend or recommending a movie during the weekdays would not be ideal. Nevertheless, most sequential models designed for the domain of
recommendation can tell which items a user might be interested in in the future, but cannot answer the question of exactly when the items should be recommended. There are a few works that center around the problem of recommendation using point process models. However, these models are usually not personalized [26, 61, 121], or do not model user preference of items meaning they can not explain why the items should be recommended at the given time [34, 55, 103, 119]. However, these methods also have the aforementioned drawbacks exist in the state-of-the-art TPPs, namely either with strict kernel parameterizations, or are too abstract to be used to interpret evolving user-system relationship.
CHAPTER 3  
Modeling Cluster Structure in Hawkes Processes

3.1 Introduction

As briefly stated in Section 2.1, self-exciting Hawkes processes have been successfully used and validated in various application domains, such as in social networks [19,144], recommendation [34,61], and education [71,135]. Typically, Hawkes process methods model each user and their activities as an independent process, assuming no relationship among them. However, in many real-world applications, this approach is not ideal for two reasons: First, user relationships, such as similarities or clustering structure presented in users and items, are not taken into account in the modeling, limiting models’ ability to generate personalized predictions. Furthermore, it requires historical observations to be available before learning, which means it cannot deal with item-user pairs that lack historical data and thus cannot be modeled and predicted in a timely manner.

Focusing on the modeling of similarities shared among users, low-rank personalized Hawkes models have recently been proposed [34], sometimes with the help of auxiliary features to reinforce the low-rank assumption [103,104]. However, to the best of our knowledge, none of the previous Hawkes models were able to represent the cluster structure between sequences, while being able to model item-user pairs with missing history and being personalized. To fill this gap, as a potential solution to RQ1, namely the question of how to model the relationship among user event sequences, in this section, we propose a novel Relaxed Clustered Hawkes process with a Gamma prior (RCHawkes-Gamma) to model and predict user interactions with different items. The model collaboratively models all item-user pairs regardless the history is available, that automatically captures the underlying clustering structure of users.

To do this, we model each item-user pair, that is the sequence of interactions of a user with an item characterized by activity times, as a uni-variate Hawkes process. By modeling all item-user pairs, our proposed model is able to capture similarities shared among users (i.e. cluster structures) by learning a low-dimensional representation of user interaction dynamics.
(i.e. personalization). As a result, even for item-user pairs without observed history (i.e. unseen data), their parameters can be inferred based on the intrinsic data group structure, without relying on auxiliary features or historical observations.

For evaluation, we apply our model on four synthetic datasets with different data missing ratios to demonstrate the robustness of our model in dealing with data sparsity. Furthermore, we also evaluate our model on 4 real-world datasets from education and recommendation domains (2.7), focusing on the applications of student procrastination modeling in online courses and user digital footprints modeling. Our experiments in both synthetic and real-world datasets show the ability of our model to accurately capture item-user interaction dynamics, resulting in a better predictive performance of future activities comparing with the state-of-the-art baseline Hawkes models. Our further analysis based on the learned parameters which characterize user interaction dynamics in identifies meaningful clusters of procrastination-related behaviors in students, as well as meaningful cluster of user checking-in dynamics across different venue categories in NYC and Tokyo.

3.2 Proposed Method

Our goal is to model all item-user pairs with historical activities that are available, and predict their future characterized into two categories, 1) the prediction of future with entirely missing history, and 2) the prediction of future with partially observed history). Specifically, suppose that there are \( N \) items and \( M \) users in the system. Each user \( u_j \) can perform a sequence of activities the corresponds to each item \( a_i \), such that all observed history sequence can be represented as item-user pair \( (a_i, u_j) \). Activities in a sequence are presented with a timestamp that marks their arrival time.

3.2.1 Uni-Variate Hawkes Intensity Function

Formally, given a sequence of activities for an item-user pair \( (a_i, u_j) \), we model its activities’ arrival times \( X_{ji} = \{ x_{ir}^j | \tau = 1, \ldots, n_i^j \} \) by a uni-variate Hawkes process, via the intensity function of time \( t \), defined as follows [49]:

\[
\lambda(t)_{ij} = U_{ij} + A_{ij} \beta \sum_{\tau=1}^{n_{ij}} \exp(-\beta(t - x_{i,\tau}^j)),
\]

(3.1)
where $x^j_{i, \tau}$ is the $\tau$-th element in the vector $X^j_i \in \mathbb{R}^{n_{ij}}$, which denotes the arrival time of the $\tau$-th activity that belongs to item-user pair $(a_i, u_j)$; $n_{ij}$ is the total number of observed activities for $(a_i, u_j)$; $U \in \mathbb{R}^{N \times M}$ is the non-negative base rate matrix, where $U_{ij}$ quantifies the expected number of activities that are triggered externally within $(a_i, u_j)$; $A \in \mathbb{R}^{N \times M}$ is the non-negative self-excitement matrix, with $A_{ij}$ representing the self-exciting or bursty nature of $(a_i, u_j)$, i.e., the expected number of activities that are triggered by the past activities; and $\beta$ is a global decay rate that represents how fast the historical activities stop affecting the future activities. Note that some entries of $A$ and $U$ can be missing if their corresponding item-user pairs do not have observed historical activities.

### 3.2.2 Relaxed Clustered Hawkes

In this proposal, we assume that user dynamics characterized by their self-exciting patterns are similar among some users, but less similar to some others, i.e. parameter matrix $A$ exhibits cluster structure column-wise. In other words, we assume that the self-excitement matrix $A$ exhibits cluster structure among users, where the similarity within a cluster is stronger than between cluster similarities. Particularly, we assume that users form $k$ clusters according to their behaviors towards all items represented in $A$’s column vectors. To impose this, we add a clustering constraint to our model using the sum of squared error (SSE) penalty, similar to K-means clustering:

$$P(A, W) = \rho_1 \text{tr}(A^\top A - W^\top A W) + \rho_2 \text{tr}(A^\top A)$$

$$= \text{tr}(A((1 + \frac{\rho_1}{\rho_2})I - WW^\top)A^\top),$$

where $\rho_1$ and $\rho_2$ are regularization coefficients; $W \in \mathbb{R}^{M \times k}$ is an orthogonal cluster indicator matrix, with $W_{ij} = \frac{1}{\sqrt{n_j}}$ if $i$ is in $j$-th cluster, and 0 otherwise (showing which users belong to which cluster); and $n_j$ is the size of cluster $j$.

Since this strict constraint is non-convex, we follow Jacob et al.’s work [58] to obtain its convex relaxation problem:

$$\min \mathcal{L}_c(A, Z) = \min \frac{\rho_2(\rho_2 + \rho_1)}{\rho_1} \text{tr}(A(\frac{\rho_1}{\rho_2} I + Z)^{-1}A^\top)$$

$$s.t. \text{tr}(Z) = k, Z \preceq I, Z \in S^M_+.$$
\(Z = WW^T \in \mathbb{R}^{M \times M}\) represents cluster-based similarity of users, with \(W\) defined in Eq. 3.2. Here, the trace norm is a surrogate of the original assumption that there are \(k\) clusters and the other two constraints are the relaxation of \(W\) being orthogonal. As a result, this equation is jointly convex to both \(A\) and \(Z\).

### 3.2.3 A mixture Gamma prior

To improve our model’s robustness to potential outliers and to possibly reduce over-fitting, we add a mixture Gamma prior on the self-excitement matrix \(A\). As a result, the summation of the first three terms in Eq. 3.8 is the A-Posteriori estimation, which not only is more robust compared with Maximum Likelihood Estimation, also provides an interpretation of each component’s hyperparameters in user clusters: i.e. the pseudo counts of externally and internally excited activities. Specifically, consider the prior for \(A_{ij}\) when user \(i\) is in \(m\)-th cluster:

\[
p(A_{ij}; \Theta_m) = \frac{1}{\Gamma(s_m)\theta_m^{s_m}}A_{ij}^{s_m-1}\exp\left(-\frac{A_{ij}}{\theta_m^{s_m}}\right),
\]

where \(\Theta_m = (s_m, k_m)\), are hyperparameters which respectively control the shape and the scale of the gamma distribution in cluster \(m\). The loss brought by the mixture Gamma prior can be computed as follows:

\[
\mathcal{L}_g = \log p(A; \Theta_1, .., \Theta_k)
\]

\[
= \sum_{X_i^j \in \mathcal{O}} \left[ \log \sum_{m=1}^k \frac{1}{k\Gamma(s_m)\theta_m^{s_m}}A_{ij}^{s_m-1}\exp\left(-\frac{A_{ij}}{\theta_m^{s_m}}\right) \right],
\]

where \(\mathcal{O}\) is the collection of all observed \(X_i^j\).

### 3.2.4 Objective Function

For our model, we need to consider the multiple sequences (as in Eq. 3.1) and add the introduced constraints. Here we first introduce a recursive function \(R\) that can be computed recursively to avoid triple summations. It is set to be 0 when \(\tau\) is 1 and updated recursively.
via the following equation.

$$ R_{ij}(\tau) = (1 + R_{ij}(\tau - 1)) \exp(-\beta(x_{i,\tau}^j - x_{i,\tau-1}^j)). \quad (3.6) $$

We also construct the matrix $T$ as follows to avoid repetitive computation in iterations:

$$ T = \left[ \sum_{\tau=1}^{n_{ij}} \left( \exp(-\beta(x_{i,n_{ij}}^j - x_{i,\tau}^j)) - 1 \right) \right]_{N \times M}. \quad (3.7) $$

To this end, the final objective function of our proposed model, given the observed activities for all item-user pairs $X$ can be described as in Eq. 3.8.

$$ \min_{A \geq 0, U \geq 0, Z} -L(X; A, U) \quad (3.8) $$

$$ = - \sum_{X_i \in O} \sum_{\tau=1}^{n_{ij}} \log \left( U_{ij} + A_{ij} \beta R_{ij}(\tau) \right) + U_{ij} x_{i,n_{ij}}^j $$

$$ + A \circ T - L_g(A; \Theta_1, \ldots, \Theta_k) + L_c(A, Z) + \rho_3 tr(A). $$

s.t. $A \geq 0, U \geq 0,$

where $L_c$ and $L_g$ are the previously defined losses introduced by clustering and gamma prior respectively and $\rho_3$ is a regularization coefficient. The trace norm regularization, is a convex surrogate for computing rank of $A$, which enables the knowledge transfer from the processes with observations to the unseen item-user pairs that do not have any observed historical activities. Finally, to not violate the definition of Hawkes process, we have non-negative constraints on $A$ and $U$.

### 3.2.5 Optimization

To solve the minimization problem in Eq. 3.8, we could use the Stochastic Gradient Descent algorithms. However, the non-negative constraints on $A$ and $U$ along with the non-smoothed trace norms can complicate the optimization. To tackle this problem, we used the Accelerated Gradient Method [88]. The key component of using this method is to compute the following proximal operator:
\[
\min_{A_z, U_z, Z_z} \|A_z - A_s\|_F^2 + \|U_z - U_s\|_F^2 + \|Z_z - Z_s\|_F^2 + (3.9)
\]

s.t. \(\text{tr}(Z_z) = k, \text{tr}(A_z) \leq c, A_z \geq 0, U_z \geq 0, Z_z \preceq I, Z_z \in S_+^M\)

where subscripts \(z\) and \(s\) respectively represents the corresponding parameter value at the current iteration and search point \([88]\). We present Algorithm 1 to efficiently solve the objective function using the Accelerated Gradient Descent framework.

### Algorithm 1: Accelerated PGA

**Input:** \(\eta > 1\), step size \(\gamma_0, \rho_3\), MaxIter

1. initialization: \(A_1 = A_0; U_1 = U_0; Z_1 = \frac{k}{M} \times I; \alpha_0 = 0; \alpha_1 = 1;\)

2. for \(i = 1\) to MaxIter do
   3. \(a_i = \frac{\alpha_{i-1}}{\alpha_i};\)
   4. \(S_i^A = A_i + a_i(A_i - A_{i-1});\)
   5. \(S_i^U = U_i + a_i(U_i - U_{i-1});\)
   6. \(S_i^Z = Z_i + a_i(Z_i - Z_{i-1});\)
   7. while True do
   8.     Compute \(A_* = \mathcal{M}_{S_i^A, \gamma_i}(A)\)
   9.     \(= (\text{TrPro}(S_i^A - \nabla \mathcal{L}(A)/\gamma_i, \rho_3))^+;\)
   10.    Compute \(U_* = \mathcal{M}_{S_i^U, \gamma_i}(U);\)
   11.    Eigen-decompose \(S_i^Z = Q\Sigma Q^{-1};\)
   12.    Compute argmin \(\sum \sigma_i - \hat{\sigma}_i)^2, \sum \sigma_i = k, 0 \leq \sigma_i \leq 1;\)
   13.    Compute \(\Sigma_* = \text{diag}(\sigma_*^1, ..., \sigma_*^M);\)
   14.    Compute \(Z_* = Q\Sigma_* Q^{-1};\)
   15.    if \(\mathcal{L}(A_*, U_*, Z_*) \leq \mathcal{L}(S_i^A, S_i^U, Z_i) + \sum_{x \in \{A, U, Z\}} \langle S_i^x, \delta \mathcal{L}(S_i^x) \rangle + \alpha_k/2\|S_i^x - x_*\|_F^2\)
   16.        then break;
   17.        else \(\gamma_i = \gamma_{i-1} \times \eta;\)
   18.        end
   19.    end
   20.    \(A_{i+1} = A_*; U_{i+1} = U_*; Z_{i+1} = Z_*;\)
   21.    if stopping criterion satisfied then
   22.        break;
   23.    else \(a_i = \frac{1 + \sqrt{1 + 4\alpha_i^2}}{2}\)
   24.    end
   25.  end
   26. end

**Output:** \(A = A_{i+1}, U = U_{i+1}, Z = Z_{i+1}\)
Algorithm Walk-through  In the following, we provide some details of Algorithm 1 shown as below. Specifically, the following subroutine is repeated in the algorithm: (1) Computation of $A_*$ (lines 8-9): The objective of this part is defined as follow:

$$\min_{A_z} F_A(A_z) := \|A_z - A_s\|_F^2 \text{ s.t. } tr(A_z) \leq c, A_z \geq 0. \tag{3.10}$$

by following the Accelerated Gradient Method schema, we compute $A_* = \mathcal{M}_{\gamma,S^A}$ (line 8), where $\mathcal{M}_{\gamma,S^A} := \frac{1}{\gamma} \|A - (S^A - \frac{1}{\gamma} \nabla \mathcal{L}(A))_+\|_F^2 + \rho_3 tr(A)$ [60]; where $S^A$ is current search point; $\gamma$ is the step size; and $\rho_3$ is the regularization coefficient. Specifically, we use trace norm projection (TrPro) [14] to solve the above minimization problem. Finally $(\cdot)_+$ projects negative values to 0 as we constraint $A$ to be nonnegative.

(2) Computation of $U_*$ (line 10): similarly to the computation of $A_*$, we compute optimal value of $U$, $U_* = \mathcal{M}_{S^U,\gamma(U)}(U)$, where $S^U$ is the current search point of $U$, and $(\cdot)_+$ is the nonnegative projection. Specifically the objective of this computation is:

$$\min_{U_z} F_U(U_z) := \|U_z - U_s\|_F^2 \text{ s.t. } U_z \geq 0. \tag{3.11}$$

(3) Computation of $Z_*$ (lines 11-14): as the constraints on $Z$ are more complicated, the proximal operator also has more terms. Specifically, the goal is to solve the following optimization problem:

$$\min_{Z_z} \|Z_z - \hat{Z}_s\|_F^2, \text{ s.t. } tr(Z_z) = k, Z_z \preceq I, Z_z \in S^M_+ \tag{3.12}$$

To solve this problem, we apply eigen decomposition on $Z_i$ such that $Z_i = Q \Sigma Q'$, where $\Sigma = diag(\hat{\sigma}_1, \ldots, \hat{\sigma}_M)$. It has been shown that $Z_* = Q \Sigma_* Q'$, where $\Sigma_* = diag(\sigma_1^*, \ldots, \sigma_k^*)$, and $\sigma_i^*$ is the optimal solution to the problem [140]:

$$\min_{\Sigma} \|\Sigma_* - \Sigma\|_F^2, \text{ s.t. } \sum_{i=1}^{M} \sigma_i = k, 0 \leq \sigma_i \leq 1. \tag{3.13}$$

To solve Eq. 3.13 with constraints, we apply the linear algorithm proposed in [68].

Remark: we want to quickly show that by solving problem 3.13, the resulting $Q \Sigma_* Q'$ provides a closed-form solution to Eq. 3.12. If denote eigen-decomposition of $M_z = \mathcal{P} \Lambda \mathcal{P}'$, by
definition, \( P'P = PP' = I \) and \( \Lambda = diag(\lambda_1, ..., \lambda_M) \) where \( \lambda_i \) for \( i = 1, ..., M \) are eigenvalues of \( z \). Then Eq. 3.12 can be equivalently written as:

\[
\min_{\Lambda, P} \|Q'P\Lambda P'Q - \Sigma\|_F^2 \quad \text{s.t.} \quad tr(\Lambda) = k
\]

\[
\lambda = diag(\lambda_1, ..., \lambda_M), 0 \leq \lambda_i \leq 1, P'P = PP' = I.
\]

It is easy to see that the constraints of the two equations with respect to \( \Lambda \) and \( \Sigma \) are equivalent. Furthermore, if denote the objectives of Eq. 3.13 and Eq. 3.14 as \( f(\cdot) \) and \( g(\cdot) \) respectively, by definition, the feasible domain of Eq. 3.13 is a subset of the feasible domain of Eq. 3.14, therefore \( f(\Sigma_\ast) \geq g(Q'P_\ast \lambda_\ast P'_\ast Q) \). On the other hand, knowing that \( \Sigma \) is a diagonal matrix, \( \|Q'P_\ast \lambda_\ast P'_\ast Q - \Sigma\|_F^2 \geq \|(Q'P_\ast \lambda_\ast P'_\ast Q) \circ I - \Sigma\|_F^2 \), meaning that the optimal objective value of Eq. 3.13 is no greater than the optimal objective value of Eq. 3.14. Therefore, the two problems are equivalent.

**Complexity Analysis**  Recall that we consider the setting where there are \( M \) users and \( N \) items. The complexity of the computation of \( A_\ast \) (line 8 – 9) is \( \mathcal{O}(MN^2) \) where a truncated SVD is used. To solve Eq. 3.12, we first apply eigen-decomposition on the \( M \times M \) matrix \( S_i^Z \) (in line 11), which has time complexity of \( \mathcal{O}(M^3) \), then we solve Eq. 3.13 which has shown to be the closed-form solution to Eq. 3.12 (line 12), a complexity of \( \mathcal{O}(M) \) can be achieved [68]. As we introduce recursive function \( R \) in Eq. 3.6, the complexity of computing loss \( \mathcal{L} \) (line 15) is \( \mathcal{O}(MNK) \) if let \( K \) denote the number of activities of the longest item-user pair. Each line of the other parts of the algorithm requires \( \mathcal{O}(MN) \) as only basic operations are involved. As a result, the time complexity per time step is \( \mathcal{O}(\max(M, N)^2M + MNK) \). In the cases where conventional Hawkes model is used, without the help of recursive function \( R \), computing the loss per time step needs \( \mathcal{O}(MNK^2) \). Note that without operations such as truncated SVD, even though a complexity of \( \mathcal{O}(MN^2) \) can be avoided for conventional Hawkes models, the parameters of item-user pairs that do not have observed activities can not be inferred.

When it comes to the number of parameters to be learned, for our model, due to our low rank and cluster structure assumption on \( A \in \mathbb{R}^{N \times M} \), the number of parameters it requires to meet these two assumptions is \( (M + N)c + 2 Mk \) where \( c < \min(M, N) \) and \( k < M \) is respectively the rank of \( A \) and the number of clusters among users, i.e. the rank.
of \( Z \in \mathbb{R}^{M \times M} \). For conventional Hawkes models, each item-user pair needs to be learned independently. As a result, the number of parameters need to complete matrix \( A \) is \( M \times N \).

**Convergence Analysis**  As mentioned earlier in this section, we have shown that Algorithm 1 repeatedly solves the subroutines respectively defined in Eq. 3.10, 3.11 and 3.12, where solving Eq. 3.12 is mathematically equivalent to solving Eq. 3.13. As it is known that accelerated gradient descent can achieve the optimal convergence rate of \( \mathcal{O}(1/k^2) \) when the objective function is smooth, and only the subroutine of solving Eq. 3.10 involves non-smooth trace norm, the focus of the following section is to provide a convergence analysis on this subroutine. Specifically, by following the outline of proof provided in Ji and Ye’s work [60], we show that a rate of \( \mathcal{O}(1/\epsilon^2) \) can be achieved in solving Eq. 3.10, even with the presence of trace norm in the objective. Specifically, if let \( A_* \) denotes the optimal solution, by applying Lemma 3.1 from Ji and Ye’s work, we can obtain the following:

\[
F_A(A_*) - F_A(A_1) \geq \frac{\gamma_1}{2} \|A_1 - S^A_1\|^2 + \gamma_1 \langle S^A_1 - A_*, A_1 - S^A_1 \rangle \tag{3.15}
\]

which is equivalent to:

\[
\frac{2}{\gamma_1} (F_A(A_1) - F_A(A_*)) \leq \|S^A_1 - A_*\|^2 - \|A_1 - A_*\|^2. \tag{3.16}
\]

Then by following the proof of Theorem 4 in Ji and Ye’s work, we can obtain the following inequality, using the equality \( \alpha^2_i = \alpha^2_{i+1} - \alpha_{i+1} \) derived from the equation in line 24 of our algorithm and the definition of \( S^A_i \) in line 4:

\[
\frac{2}{\gamma_{i+1}} [\alpha^2_i (F_A(A_i) - F_A(A_*)) - \alpha^2_{i+1} ((F_A(A_{i+1}) - F_A(A_*)))] \tag{3.17}
\]

\[
\geq \|\alpha_{i+1}A_{i+1} - (\alpha_{i+1})A_i - A_*\|^2 - \|\alpha_i A_i - (\alpha_i - 1)A_{i-1} - A_*\|^2.
\]

As \( \eta \geq 1 \) and we update \( \gamma_{i+1} \) by multiplying \( \eta \) with \( \gamma_i \), we know that \( \gamma_{i+1} \geq \gamma_i \). By
plugging in this inequality to Eq. 3.17, we can obtain the following:

\[
\frac{2}{\gamma_i} \alpha_i^2 (F_A(A_i) - F_A(A_\ast)) - \frac{2}{\gamma_{i+1}} \alpha_{i+1}^2 (F_A(A_{i+1}) - F_A(A_\ast)) \\
\geq \|\alpha_{i+1} A_{i+1} - (\alpha_{i+1} - 1) A_i - A_\ast\|^2 - \|\alpha_i A_i \\
- (\alpha_i - 1) A_{i-1} - A_\ast\|^2.
\]

By summing up each side of Eq. 3.18 from \(i = 1\) to \(i = k\), then combining with Eq. 3.16, we can obtain the following:

\[
\frac{2}{\gamma_i} \alpha_i^2 (F_A(A_i) - F_A(A_\ast)) \leq \|A_1 - A_\ast\|^2 \\
- \|\alpha_i A_i - (\alpha_i - 1) A_{i-1} - A_\ast\|^2 + \frac{2}{\gamma_1} (F_A(A_1) - F_A(A_\ast)) \\
\leq \|A_i - A_\ast\|^2 - \|\alpha_i A_i - (\alpha_i - 1) A_{i-1} - A_\ast\|^2 \\
+ \|A_0 - A_\ast\|^2 - \|A_i - A_\ast\|^2 \\
\leq \|A_0 - A_\ast\|^2
\]

Using the fact that \(\alpha_i \geq \frac{i+1}{2}\) (can be shown using induction from line 24 of the algorithm), we can obtain:

\[
F_A(A_i) - F_A(A_\ast) \leq \frac{2\gamma_1 \|A_\ast - A_0\|^2}{(i + 1)^2}.
\]

### 3.3 Experimental Evaluation

In this section, we evaluate the effectiveness of RCHawkes-Gamma in modeling user interaction dynamics while capturing underlying cluster structure of the data. We conduct three sets of experiments and compare RCHawkes-Gamma with several state-of-the-art point process models on both synthetic and real-world datasets.

#### 3.3.1 Datasets

In this work, we consider both synthetic and real-world datasets, described as follows:

**Synthetic Data.** We first consider simulated data. Prior work shows that students in online courses are shown to exhibit self-exciting learning behaviors [135, 136]. Motivated by
this, we create simulated item-user pairs based on Hawkes processes. More specifically, we first construct the parameter matrices. We build $A_s$ by: 1) sampling $k = 3$ sets of column $\alpha$’s from different Gamma distributions; 2) adding white noise ($\sigma^2 = 0.1$); and 3) shuffling all columns randomly to break the order. We build $U_s$, by sampling it from a normal distribution. Then, we sample 150 activities for each item-user pair using the Ogata thinning algorithm [89], which is the most commonly used sampling method in the related literature. Finally, we obtain 5400 simulated item-user pairs and 810K synthetic activities.

**Computer Science Course on Canvas Network (CANVAS).** This real-world MOOC dataset is from the Canvas Network platform [17]. Canvas Network is an online platform that hosts various open courses in different academic disciplines. The courses offered on this platform have multiple types of learning resources, which includes learning modules, assignments/quizzes and discussions. The computer science course we use happens during $\sim 6$ weeks. In each week, an assignment-style quiz is published in the course resulting in 6 course assignments. In total, we extract $\sim 740K$ assignment-related activity timestamps from 471 students.

**Big Data in Education on Coursera (MORF).** Our second real-world dataset is collected from an 8-week “Big Data in Education” course in Coursera platform. The dataset is available through the MOOC Replication Framework (MORF) [4]. Within each week, it provides students with multiple types of learning resources such as assignments, video lectures, and discussion forums. In total, we extract $\sim 102K$ activities of 675 students related to 8 assignments.

**Foursquare User Check-ins in NYC and Tokyo.** This data contains users’ digital footprints, i.e., their check-ins at various venues for around 10 months collected from Foursquare by Yang et al. [132]. Two datasets are included, containing user check-ins in New York City (Foursquare-NYC) and Tokyo (Foursquare-TKY), and are respectively used as two datasets in this study. Users and venues that have less than 50 check-ins are excluded for this study to obtain a sufficient learning length for our Hawkes-based method. In total, from Foursquare-NYC we obtain 114 users and their $\sim 27K$ check-ins in 54 venues, with a item-user missing percentage of 95.73%. And in Foursquare-TKY, we obtain 155 users and $\sim 35K$ check-ins in 55 venues, with a item-user missing percentage of 95.21%.
3.3.2 Experiment Setup

We test our method in two scenarios according to our application: 1) when the historical observations are available, we want to predict what will happen in the future based on the history, and 2) when the whole sequence of activities for a student-assignment pair is completely missing, we want to infer its future without observing its history. To test the model’s performance in predicting the future in these two scenarios, we split our data into the following 3 sets: **training set** that contains the initial historical observations, which is used to train the model for parameter inference; **partially missing test set** that contains the rest of the historical observations, that is used for testing the first scenario. Finally, the **completely missing test set** contains the entire observations of the sequences, and it is used to examine models’ ability in generating personalized and accurate predictions for unobserved sequences, i.e. scenario 2.

To check models’ sensitivity to the sparsity of the data, in simulated data, we create sets of simulation data with different missing ratios, where we randomly select a ratio of $r = [0.1, 0.3, 0.5, 0.7]$ amount of students’ last two assignment activities to be entirely missing (i.e. completely missing set), and for the rest of the student-assignment pairs, the first 70% of activities are used for training, and the last 30% are used for testing (i.e. partially missing set). In student learning datasets, to simulate a realistic student learning scenario, we use activities that took place before the mid point of the last assignment as training. In user check-in datasets, follow the convention, 20% of the user-item pairs that have observations are randomly masked (i.e. completely missing set), among the rest of the user-item pairs, the first 80% of the activities are used for training, and the rest is used for testing (i.e. partially missing set).

3.3.3 Baseline Approaches

We consider two sets of state-of-the-art baselines: the ones that are able to infer unseen data, and the ones that cannot. A summary of all baseline approaches is presented in Table 3.1. In the following, we briefly introduce each of the baselines.

**EdMPH**: A Hawkes model that was recently proposed to model student procrastination in Educational Data Mining domain [135]. It applies a Multivariate Hawkes Model which utilizes student activity types as extra information, and cannot infer unseen data.
**RMTPP**: A Recurrent Neural Network Hawkes model to represent item-user interactions [32]. It does not directly infer parameters of unseen data and it uses activity markers as an input.

**ERPP**: A similar approach to baseline RMTPP, but it includes time-series loss in the loss function [129].

**HRPF** and **DHPR**: Two Poisson factorization models proposed in [55] that do not require user-network as auxiliary features. These models, however, do not directly model the time-dependencies between the future and the past, thus cannot quantify activity self-excitement.

**HPLR**: An item recommendation model using Hawkes process [34]. It is the most similar to ours, as it imposes a low rank assumption on matrices $A$ and $U$ and can infer unseen data. However, unlike our model, it does not consider the cluster structure of parameter matrix $A$.

**RCHawkes**: A variation of our proposed model that does not use a Gamma prior. Its objective is to find the maximum of log-likelihood rather than the maximum of A-posterior.

<table>
<thead>
<tr>
<th>Application</th>
<th>Model</th>
<th>Infer Unseen Data</th>
<th>Require No External Features</th>
<th>Model Time Dependency</th>
</tr>
</thead>
<tbody>
<tr>
<td>EDM</td>
<td>RCHawkes-Gamma</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>RChawkes</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>EdMHP</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Rec-Sys</td>
<td>HPLR</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>ERPP</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>HRPF</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td></td>
<td>DRPF</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>RMTPP</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table 3.1**: A summary of baseline approaches.

### 3.3.4 Experimental Evaluation I: Estimated Parameters on Simulated Data

In the simulated dataset, as we know the true parameters (i.e. $A$ and $U$), we compare the Root Mean Squared Error (RMSE) of estimated $\hat{A}$ and $\hat{U}$, varying missing ratio $r$ \(^3\). The results are presented in Tbl. 3.2.

RCHawkes-Gamma and RCHawkes outperform the baseline methods usually by a large

---

\(^3\)Baseline ERPP, RTMPP, and EdHawkes cannot be used in this analysis, since they parameterize the processes differently.
margin, for both the sequences with seen and unseen history. Also, even though all models perform worse with the increase of $r$, RCHawkes-Gamma and RCHakwes’ RMSEs have a lower standard deviation, indicating less variation in their performance even in high missing data ratios. Additionally, the models’ performances in completely missing test set are generally worse than their performances in the processes with observed historical activities. One possible reason is that the Hawkes parameters for completely missing test set in this simulation can only be inferred from the similar processes with observed data by leveraging the row and column relatedness, while the true characteristics of the unseen processes can not be entirely captured as there are no observations that the models can use.

In short, by modeling all sequences jointly using the proposed approach, our model is able to recover the true dynamic of data more accurately and robustly compared with baseline approaches.

### 3.3.5 Experimental Evaluation II: Clustering Structure of Hawkes Parameters

To see if the cluster structure of users is well captured by each model, we compute and present the correlation matrix of $\hat{A}$ between simulated users with the recovered cluster orders in Figure 3.1.

As it is shown in this figure, our proposed model recovers this structure closer to the ground truth (Figure 3.1 (a)), i.e. a higher correlation within clusters (darker blocks) and a lower correlation between clusters (lighter blocks). HPLR introduces unnecessary correlations between clusters, possibly because of not having the student cluster assumption. HRPF simply assumes all assignment-student pairs share the same parameter, thus has a meaningless correlation of 1 among all students. Finally, although DRPF improves HRPF by considering activity self-excitements, it fails to capture any meaningful correlation within
Figure 3.1: The ground truth of $A$’s correlation matrix (a), and the estimated $\hat{A}$’s correlation matrix learned by each model.

3.3.6 Experimental Evaluation III: Returning Time Prediction

To evaluate model time prediction performance, we use a metric that has been commonly used in the literature of Hawkes-based models, i.e. RMSE on Time Prediction (TP) that is defined on the estimated next activity time, given the observed history. The baselines that do not directly infer parameters on completely missing test set (the future assignment scenario) are not included in the completely missing test set evaluation. Following the method used in [34], we sample future activities based on the learned parameters via Ogata’s thinning algorithm. More specifically, we first use Ogatha thinning algorithm to sample inter-arrival times $\Delta t_z$, which is the time difference between $z$-th and $(z - 1)$-th
events. Then, we compute the predicted time of \( z \)-th event \( \hat{x}_z \) as \( \hat{x}_z = \hat{x}_{z-1} + \frac{1}{N_t} \sum_{i=1}^{N_t} \Delta t^i_z \), where \( N_t \) is the trail number for the sampling and \( \Delta t^i_z \) is the sampled inter-arrival time at the \( i \)-th trail. The intuition is that inter-arrival times are sampled \( N_t \) times, then the sample mean of all \( N_t \) trails is used as the approximation of the actual inter-arrival time.

**Figure 3.2:** Time prediction error and 95% confidence interval on synthetic partially missing test set (left) and completely missing test set (right) datasets with varying data missing ratios \( (r) \)

**Figure 3.3:** Time prediction error in partially missing test set (left) and completely missing test set (right) data with 95% confidence interval respectively on student learning datasets (upper) and user check-in (lower) datasets.

Fig. 3.2 and 3.3 respectively present the prediction error and 95% confidence interval on simulated and real-world data. As we can see from Figure 3.2, the proposed methods
RCHawkes-Gamma and RCHakwes consistently outperform other baselines in all settings, except when the missing ratio is 0.1. In that case, RMTPP and ERPP achieve the smallest error in partially missing test set. However, unlike the proposed models that are almost invariant to the increase of the missing ratio, ERPP and RMTPP’s performances change dramatically with increasing $r$. More importantly, they lack the ability to directly predict the next activity time when the activity history is unseen. We can also see that HRLR has comparable performance to the proposed method in synthetic data. On the other hand, HRPF and DRPF have the highest prediction errors, possibly because they do not directly model self-excitement of activities. In real-world datasets, it is shown in Fig. 3.2 that the proposed approach RCHawkes-Gamma and variant RCHawkes usually outperform baselines by big margins, and all models’ performances are almost consistent to the synthetic data. On the other hand, even though EdMHP and RMTPP achieve competitive performances, they lack the ability to directly predict the next activity time when the activity history is unseen. It is worth noting though that the advantage of the proposed approach is highlighted in user check-in datasets where the data sparsity is particularly high ($\sim 95\%$), further showing the robustness of the proposed model to high missing ratio.

In summary, by modeling all sequences jointly while reinforcing the learning of group structure of the data, our proposed model is able to accurately represents point process dynamics (Chp. 3.3.4), successfully recover the underlying group structure of the data (Chp. 3.3.5), and generating more accurate time predictions (Chp. 3.3.6) compared with the state-of-the-art TPPs.

### 3.4 Understanding User Activity using RCHawkes-Gamma

To better understand users and their activity dynamics, in this section, we provide the analysis of user interaction dynamics respectively in the student learning (Ch. 3.4.1) and user check-in (Ch. 3.4.2) datasets, utilizing the underlying group structure of item-user pairs mined by the proposed model.
3.4.1 Studying Student Procrastination Behavior

In the following section, we will focus to finding the connections between the characteristics of students’ learning activities (parameterized by our model) and students’ cramming behaviors.

Since procrastination does not have a quantitative definition, in the first step of our analysis, we define the following measure to describe the degree of student procrastination presented in MOOCs: delay = \( \frac{t_{a_{ij}} - t_{s_{ij}}}{t_{d_{ij}} - t_{s_{ij}}} \) to quantify student j’s normalized delay in starting any activity that is associated with assignment i, where superscript s, a, d respectively represents the start of the assignment, the first, and the last activity in the student-assignment pair. Intuitively, this measure is the absolute time that student j delays in starting assignment i, normalized by the duration that assignment i is available for student j. Note that this measure is just a simple representation and cannot replace our model in predicting next activity times or uncovering cluster structures.

In order to show how students activities parameterized by Hawkes and student delays are associated, we compute the Spearman’s rank correlation coefficient between each pair of the variables. As we can see in Table 3.3, the two-sided p-values suggest that the correlations between these variables are statistically significant. We can also see that all the correlation coefficients are positive, meaning that student delays are positively associated with the base rate, i.e. expected number of occurrences per unit time that are excited by external stimuli (for example deadlines), and the burstiness of the occurrences. On the other hand, by looking at the two courses side-by-side, we can see that delay is more strongly associated with \( \alpha \) in CANVAS. But, its association with the base rate \( \mu \) is stronger in MORF. This suggests two different kinds of relationships between students and assignments: while in CANVAS big bursts of activities might suggest delays, in MORF small but frequent activities are associated with student delays.

In the clustering analysis, students are clustered via K-means clustering algorithm, which has a similar objective to the cluster-structure term in our model (Eq. 3.2), on the learned \( \hat{A} \) matrix. Specifically, student \( u_j \) is represented by the vector of estimated self-excitement parameters \( (\hat{\alpha}_{1,i}, ..., \hat{\alpha}_{N,i}) \) that are learned by RCHawkes-Gamma, and the cluster number for K-means is decided via grid search by looking at SSE. To examine the possible differences between clusters of students in terms of student delays, we conduct the Kruskal-
self-excitement: $\alpha$  
base rate: $\mu$  
<table>
<thead>
<tr>
<th>delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.3: Spearman’s correlation between learned parameters and computed normalized student delays. p<0.001*** p<0.01 ** p<0.05*

Wallis test on all student delays across the clusters for each assignment. We report the average delay of all students in each cluster and for each assignment. The results are shown in Table 3.4 for CANVAS and in Table 3.5 for MORF dataset. In CANVAS, 4 student clusters

<table>
<thead>
<tr>
<th>Assign. #.</th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
<th>cluster 4</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>81</td>
<td>144</td>
<td>207</td>
<td>39</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.3335</td>
<td>0.4583</td>
<td>0.6108</td>
<td>0.9064</td>
<td>1.34E-16***</td>
</tr>
<tr>
<td>2</td>
<td>0.6245</td>
<td>0.5788</td>
<td>0.8476</td>
<td>1.0854</td>
<td>3.59E-09***</td>
</tr>
<tr>
<td>3</td>
<td>0.6911</td>
<td>0.7143</td>
<td>0.8633</td>
<td>0.9655</td>
<td>4.36E-05***</td>
</tr>
<tr>
<td>4</td>
<td>0.6050</td>
<td>0.6958</td>
<td>0.8515</td>
<td>1.0717</td>
<td>0.0008***</td>
</tr>
<tr>
<td>5</td>
<td>0.5969</td>
<td>0.7080</td>
<td>0.9084</td>
<td>1.1217</td>
<td>0.0195*</td>
</tr>
<tr>
<td>6</td>
<td>0.5351</td>
<td>0.7647</td>
<td>0.9002</td>
<td>1.0970</td>
<td>0.0149*</td>
</tr>
</tbody>
</table>

Table 3.4: Kruskal Wallis test on delays in different clusters in CANVAS dataset. p<0.001*** p<0.01 ** p<0.05*

<table>
<thead>
<tr>
<th>Assign. #.</th>
<th>cluster 1</th>
<th>cluster 2</th>
<th>cluster 3</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>573</td>
<td>34</td>
<td>68</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.4991</td>
<td>0.6710</td>
<td>0.4477</td>
<td>2.30E-09***</td>
</tr>
<tr>
<td>2</td>
<td>0.5120</td>
<td>0.7288</td>
<td>0.4855</td>
<td>1.90E-08***</td>
</tr>
<tr>
<td>3</td>
<td>0.5570</td>
<td>0.6904</td>
<td>0.6105</td>
<td>7.50E-05***</td>
</tr>
<tr>
<td>4</td>
<td>0.4699</td>
<td>0.6122</td>
<td>0.5360</td>
<td>0.0004***</td>
</tr>
<tr>
<td>5</td>
<td>0.5626</td>
<td>0.6358</td>
<td>0.6308</td>
<td>0.0070***</td>
</tr>
<tr>
<td>6</td>
<td>0.5329</td>
<td>0.6236</td>
<td>0.6642</td>
<td>8.56E-06***</td>
</tr>
<tr>
<td>7</td>
<td>0.4325</td>
<td>0.5598</td>
<td>0.7672</td>
<td>2.12E-20***</td>
</tr>
<tr>
<td>8</td>
<td>0.3974</td>
<td>0.5172</td>
<td>0.7629</td>
<td>3.84E-27***</td>
</tr>
</tbody>
</table>

Table 3.5: Kruskal Wallis test on delays in different clusters in MORF dataset. p<0.001*** p<0.01 ** p<0.05*
are found. These clusters all have significant differences in terms of delays. For example, students in cluster 1 have the smallest delay, with a general decreasing trend towards the later assignments. On the other hand, delays are the worst for students in cluster 4, with an average delay greater or close to 1 for all assignments, which implies that this group of students tend to start the assignments very close to or even later than the deadline. In the 3 clusters that are found in the MORF dataset, the p-values of Kruskal-Wallis tests show strong evidence of cluster differences for each assignment. Specifically, the majority of the students in the MORF course are in cluster 1 and their delays are overall the lowest comparing to the other two clusters. They tend to delay less and less over time. On the other hand, students in cluster 3 start the course with a low delay but increase their delay so fast that at the end of the course, they turn out to be the students who delay the most. This analysis demonstrates that the self-excitement parameters have strong associations with student delays, which not only reinforces the findings from the correlation analysis, but also suggests that they are good indicators in characterizing students’ cramming behaviors.

3.4.2 Studying User Checking-in Behavior

In foursquare datasets, user-related information was not made available for privacy concerns. But instead, we have the categories of the venues where user has checked-in. In this analysis, similar to the previous study of student procrastination modeling, we cluster the venues using the learned parameter $\hat{A}$. Specifically, each venue $a_j$ is represented by the vector of self-excitement parameters $(\alpha_{1,j}, ..., \alpha_{M,j})$ estimated by RCHawkes-Gamma. The summary of the findings in Foursquare-NYC and Foursquare-TKY are respectively given in Table 3.6 and Table 3.7.

As it is shown in the tables, three meaningful clusters of venue categories in both Foursquare-NYC and Foursquare-TKY are identified. For example, in Foursquare-NYC dataset, cluster 1 discovered based on learned parameters are shown to be mostly related to restaurants (e.g. Chinese or American restaurants) and entertainment venues (bar or theater). Cluster 2 is shown to be associated with fast food or daily commute, including examples such as food truck, coffee shop, home, and office. Cluster 3 on the other hand are generally infrastructures such as road, bridge or airport. By looking at the means of the learned self-excitement parameter $\alpha$ and base rate parameter $\mu$ in all 3 clusters, we see
that in cluster 3 (i.e. infrastructure), user check-ins are shown to have high base rate and high self-excitement. This observation agrees with common sense that these venues are the heart of the cities where people rely on them on a frequent basis. On the other hand, venues such as restaurants and entertainment (i.e. cluster 1) are shown to have smaller base rate
and self-excitement. A possible explanation is that these venues generally have their busiest peak times after working hours, therefore have less frequent and bursty check-in dynamics. Similar clusters of venues and user checking-in dynamics characterized by self-excitement and base rate are also found in Foursquare-TKY. For example, food and entertainment venues with lower self-excitement and base rate are identified, whereas a cluster of venues mostly associated with commute and transportation are shown to have higher base rates and self-excitements, possibly explained by the frequent usage of these venues such as bus station or subway.

To understand user checking-in dynamic in a finer-grained manner, we provide the distributions of users’ local check-ins times during a day across 7 days a week. This visualization is provided in Fig. 3.4 and Fig. 3.5 for Foursquare-NYC and Foursquare-TKY respectively.

![Box plot of user check-in times in Foursquare-NYC.](image)

**Figure 3.4: Box plot of user check-in times in Foursquare-NYC.**

By looking at clusters side by side, it is observed that entertainment venues and restaurants (blue boxes) usually are visited later in the day than the other categories, for example, comparing with commute and fast food in NYC (orange box in Foursquare-NYC) or daily life and stores in Tokyo (orange box in Foursquare-TKY). On the other hand, venue categories that user usually visited earlier during the day belong to the cluster of infrastructure or transportation (green boxes in Foursquare-NYC and Foursquare-TKY). Furthermore, it
is also interesting to compare clustered user check-ins between weekdays and weekends. Particularly, in NYC, check-ins at commute, fast food and infrastructure related venues are typically distributed later during the day than week days. However, check-ins at entertainment and food venues in weekend mostly are distributed earlier than weekdays and ends later, suggesting a general longer time period for these venues. Interestingly, possibly due to the differences of living styles in New York from Tokyo, this observation is almost reversed in Tokyo. Venues that falls in the food and entertainment cluster are seemed to be checked in later with a shorter span during weekends than in weekdays, while Friday seems to be the peak of this cluster of venues (i.e. later check-in times in general).

To summarize, this analysis demonstrates that the proposed model is able to discover the underlying clustering structure of venue-item pairs and can be used to provide explanation of user activity patterns and preferences temporally.

3.5 Conclusion

In this section, to answer RQ1, we proposed a novel uni-variate clustered Hawkes process model, RCHawkes-Gamma to model user interactions with items. Particularly, the
proposed method models user interactive activities on all item-user pairs jointly and assumes cluster structures between users and relatedness between items. We test our proposed model on four synthetic dataset with different sparsities, as well as four real-world datasets from education and recommendation domains. The results of our experiments show that our proposed model is able to recover the underlying cluster structure of the data, can predict user next activity time with lower time prediction error on both seen and unseen data, across both synthetic and real-world datasets, compared to the state-of-the-art baseline approaches. We also study and analyze the parameters learned by the proposed model on both student datasets and user check-in datasets which sheds light on the understanding of user behaviors temporally. More specifically, in the student datasets with the goal to understand student procrastination-like behaviors, our analysis reveals positive associations between student procrastination and their learning dynamics characterized by our model’s parameters. The model also discovers meaningful clusters of students who show different procrastination-like behavior trends during the course. Furthermore, in the user check-in datasets, our model reveals meaningful clusters of checking-in behaviors across all venue categories that show different checking-in dynamics.
CHAPTER 4
Modeling External Stimuli in Hawkes Processes

4.1 Introduction

As introduced in Section 1.1, effectively representing external stimuli and their effects on user event sequences is important in describing and understanding the dynamics of user-system interactions. In the literature, it is common practice to model external stimuli and their effect based on the assumption that they are invariant to time, realized by using a constant function in traditional TPPs or representing them as bias vectors in neural TPPs \([34, 82, 129, 149]\). However, this assumption is very strict and can be easily violated in real-world applications. More specifically, in the applications of modeling user-system interactions, it is natural to assume that user activities can be externally affected by both users (e.g., personal preference of the interaction frequencies) and the characteristics of the system (e.g., external “deadlines” such as assignment due or “end of the sale” type of promotions). Both of the aforementioned types of stimuli can cause their own distribution of activities, and when combined, they can cause even more complex dynamics of effects. Ignoring these external stimuli not only reduces the model’s descriptive power of the data due to oversimplification, but it also ignores factors that carry important underlying characteristics of user-system interaction patterns.

Motivated by this, we focus on the student procrastination modeling problem as our first step toward effectively modeling external stimuli in user event sequences. Modern point process models used in education have primarily parameterized external triggering effects as constants, resulting in the omission of factors such as class schedules and personalized student habits. For example, as a format of external stimuli, an assignment deadline may only begin to show its triggering effect as it approaches. Personal habits of students (e.g., log-in time and frequency), which reflect their time management skills, may also evolve over time. To deal with this problem, we propose our Stimuli-Sensitive Hawkes Process (SSHSP) model \([138]\), which is designed to capture important types of external stimuli as parameterized functions of time: the effect of student’s personal study time and frequency
habits (i.e. stimuli come from the user), and the effect of the course exerts to students (i.e.
imuli come from the system), more specifically, the availability of the assignments and the
deadlines of the assignments.

4.2 Proposed Method

Similar to the problem of student procrastination modeling introduced in Sec 3.3,
consider we are given $U$ students and $N$ assignments in a course. We assume that the
time when student $u_i$ interacts with assignment $a_j$ depends on two things: (1) the effects
of external stimuli (e.g., the deadline of $a_j$ is approaching, therefore student $u_i$ starts to
review the lectures and practices on the quizzes), and (2) the self-exciting nature of the
events, in other words, past events can trigger the future ones (e.g., student $u_i$ decides to
work on assignment $a_j$ because they just watched the lecture video that is related to $a_j$).
To capture these triggering effects which can be important in explaining students behaviors
in the course, we propose to model the collection of activity timestamps of student $u_i$’s
interactions with assignment $a_j$, or student-assignment pair $(u_i, a_j)$, as a point process.
Formally, given a student-assignment pair $(u_i, a_j)$, we describe it as the timestamps of all
student $u_i$’s interactions with assignment $a_j$: $X_{ij} = \{x_{ij}^\tau | \tau = 1, ..., K_{ij}\}^4$.

4.2.1 Parameterization of External Stimuli and Self-excitement

Modeling external stimuli. We parameterize the following 3 types of external stimuli
that can trigger student’s interactions with the assignment. Firstly, the effect of student
habit: we assume that each student interacts with the course based on their own periodical
studying schedule. For example, some students habitually log in to the course at noon every
day, but some prefer to study after midnight. Secondly, the decaying effect of the assignment
availability (opening): we assume that students’ activities can be triggered once the assign-
ment is posted. However, this effect decays over time. For example, once an assignment is
posted, students may log in and check the assignment requirements or deadlines, or revisit
it later for the detailed descriptions. However, over time, this effect will die out and will be
dominated by other stimuli. Finally, the deadline of an assignment: we assume that student

\[^4\]For simplicity, without causing any confusion, we omit the individual subscripts $i$ and $j$ in the rest of
this section.
activities can be triggered by the deadline, and this effect gets stronger by approaching the deadline and wears off eventually.

Formally, we define the base rate intensity for students at each time $t$ as a combination of each of the above stimulus as in Equation 4.1.

$$
\mu(t) = \gamma^d \mu^d(t) + \gamma^o \mu^o(t) + \gamma^h \mu^h(t),
$$

(4.1)

$$
\mu^h(t) = \sin\left(\frac{2\pi}{s}(t + p)\right) + c,
$$

(4.2)

$$
\mu^o(t) = b^{t/s},
$$

(4.3)

$$
\mu^d(t) = \begin{cases} 
\frac{1}{\sqrt{2\pi}v(d-m-t/s)} \exp\left(-\frac{(\ln(d-m-t/s))^2}{v}\right) & \text{if } d-m \leq t/s, \\
0 & \text{if } d-m > t/s.
\end{cases}
$$

(4.4)

Specifically, Eq. 4.2 models the activity intensity triggered by students’ habits as a sinusoidal function. In other word, $\mu^h(t)$ captures periodicity of length $s$, that peaks at $p$. $c$ can be interpreted as the minimum number of the activities triggered by the student habits, which works as a base of $\mu^h(t)$. Eq. 4.3 models the opening effects of the assignment as an exponential function parameterized by $b$, with a decay speed of $1/b$ over time scaled by $s$. This formulation will result in an exponentially less number of activities, as a result of assignment posting, as time passes. Eq. 4.4 models the effect of deadline via a reversed log-normal function. $d$ here is the known time of the assignment deadline, $d-m$ represents the time when the deadline’s triggering effect on student activities is over. As a result, $m$ represents the difference between the end of the deadline’s effect and the deadline. If the effect of deadline is over after the actual time of deadline (e.g. late submission), $m$ would be negative. Otherwise, $m \geq 0$. Non-negative $v$ controls how intense the activities are closing to the deadline and how fast this effect decays after the peak. This formulation represents that student activity intensities will peak around their last assignment-related activity, which is close to the deadline, either before or after it. $\gamma^h$, $\gamma^o$ and $\gamma^d$ respectively are the weight coefficients that describe the importance of $\mu^h(t)$, $\mu^o(t)$ and $\mu^d(t)$.

**Modeling internal stimuli.** To model the effect of past activities, we adopt the following
conventional self-excitation function used in point processes:

\[ s(t) = \sum_{x^\tau < t} \alpha \beta e^{-\beta(t-x^\tau)}, \quad (4.5) \]

The above excitation function characterizes the effect of each historical event \( x^\tau \) to current time \( t \), as a decaying function of the time difference between \( t \) and \( x^\tau \), with the decaying speed of \( 1/\beta \). Therefore, the more recent a historical event is, the more effect it has in terms of self-excitation. \( \alpha \) can be shown to be the branching ratio under this definition, i.e. the expected number of activities that are triggered by a given activity. Thus it is called the self-exciting coefficient.

**Intensity function.** Finally, our intensity function for one student-assignment pair can be defined as follows:

\[ \lambda(t) = \mu(t) + s(t) \]

\[ = \gamma^h (\sin(\frac{2\pi}{s}(t + p)) + c) + \gamma^o b^{t/s} + \gamma^d \frac{1}{\sqrt{2\pi}v(d - m - t/s)} e^{-\frac{(\ln(d - m - t/s))^2}{v}} + \sum_{x^\tau < t} \alpha \beta e^{-\beta(t-x^\tau)}. \quad (4.6) \]

As we can see, the intensity is the combination of base rate function \( \mu(t) \) that models external stimuli, and the excitation function \( s(t) \) that models the self-excitement. The proposed intensity function falls in the category of a popular family of point process, i.e. Hawkes processes, which conventionally model the effect of all external stimuli as a constant. As our proposed model parameterizes the effects of different external stimuli in the educational setting as functions of time, we call our model Stimuli-Sensitive Hawkes process model (SSHP).

**Matrix representation for all student-assignment pairs.** Equation 4.6 above represents the intensity function for activities of one student on one assignment. To model all student activities on all assignments, one can model them as separate sequences and learn the parameters for each sequence independently. However, this kind of model will result in two limitations. Firstly, no parameters can be learned for student-assignment sequences that are completely unobserved, and thus, student activities in such sequences cannot be predicted. For example, consider a student, who has not started working on a future as-
signment by the end of the observation window, or a student, who skips an assignment for now and plans to come back to it later. Excluding these sequences from the study largely limits the capacity of the model in our application. Secondly, the parameters of the model that are not assignment-related, such as student habit parameters, are going to be learned independently for each sequence. As a result, they will lose meaning. A common approach to deal with these limitations is to extend the data collection window, which could be costly and inefficient. Another solution could be using the learned parameters from the observed sequences and applying them to the sequences that do not have observations. However, such an approach cannot provide personalized inferences, thus is not ideal. To deal with these problems, while learning personalized parameters for students, motivated by the proposed model and preliminary findings for RQ1 (Ch. 3), we again assume similarity between the learned parameters for all student-assignment pairs. Particularly, we represent the relationship between students and assignments as a student-assignment matrix, where a row is a student and each column represents an assignment from the course. We represent the student-assignment-related parameters of the model in such a matrix format, model the student-related parameters of the model in a vector format (so that they are shared between all assignments for a student), and share some generic parameters of the model between all students. As a result, for example, the intensity function of student-assignment pair \((u_i, a_j)\) can be defined as the parameters that correspond to the \(j\)-th cell in row \(i\) from the parameter matrices. More specifically, the parameters are set to follow the following three structures:

1. **Scalars**: following the convention of Hawkes processes, we set global decay coefficient \(\beta\) to be shared among all sequences. We also set \(s\) to be a global scalar, so that time \(t\) is scaled to the same unit across all student-assignment pairs.

2. **Vector sets \(\phi\)**: We let \(c = (c_1, \ldots, c_U)\), \(p = (p_1, \ldots, p_U)\), \(b = (b_1, \ldots, b_U)\) and \(v = (v_1, \ldots, v_U)\) to be vectors, assuming a student’s habit is unchanged across the assignments (i.e. \(c\) and \(p\)). Similarly, their sensitivity to the effect of assignment openings (i.e. \(b\)). Furthermore, how fast their activities becoming intense once the deadline started affecting them (i.e. \(v\)) is also set to be shared among assignments.

3. **Low-rank matrices \(\Theta\)**: For each of the rest of the parameters, we consider a matrix format and assume similarity among student-assignment pairs, i.e. a low rank structure on the matrix format.
4.2.2 Objective Function

**Maximum likelihood estimation on one sequence.** Given a student assignment pair \((u_i, a_j)\)'s historical activities \(X_{ij} = \{x_{ij}^{\tau} | \tau = 1, ..., K_{ij}\}\) over the time period \([0, T]\), we consider the parameter set \(\theta = (\alpha, p, c, b, v, m, \gamma^h, \gamma^o, \gamma^d)\). Following the convention of Hawkes process literature, we set the decay \(\beta\) as a hyperparameter. Based on the assumption that each student has the same habitual seasonal patterns towards all assignment, \(s\) that controls the scale of the sine function is also set as a hyperparameter. Then the final explicit form of log-likelihood \(l(\theta)\) can be shown as below:

\[
l(X; \theta) = \log L(\theta) = \sum_{\tau=1}^{K} \log(\lambda(x^\tau)) - \int_{0}^{x^K} \lambda(u)du
\]

\[
= \sum_{\tau} \log (\gamma^d \mu^d(x^\tau) + \gamma^o \mu^o(x^\tau) + \gamma^h \mu^h(x^\tau)) + \alpha \beta R(\tau) - \gamma^d U^{ds}(x^K) - \gamma^o U^{os}(x^K) - \gamma^h U^{hs}(x^K)
\]

\[
+ \alpha \sum_{\tau=1}^{K} (e^{-\beta(x^K-x^\tau)} - 1).
\]

\(U^{ds}(\cdot), U^{os}(\cdot), U^{hs}(\cdot)\) is respectively the cumulative intensity of \(\mu^d, \mu^o\) and \(\mu^h\) introduced due to the integral the above equation, which have the explicit forms as follows:

\[
U^{hs}(x^\tau) = \int_{0}^{x^\tau} \mu^h(u)du
\]

\[
= \frac{1}{\pi} \left(-24s \cos\left(\frac{\pi x^\tau + \pi ps}{24s}\right) + 24s \cos\frac{\pi p}{24} + \pi cx^\tau\right),
\]

\[
U^{os}(x^\tau) = \int_{0}^{x^\tau} \mu^o(u)du = s\left(\frac{b^{x^\tau/s} - 1}{\ln b}\right),
\]

\[
U^{ds}(x^\tau) = \int_{0}^{x^\tau} \mu^d(u)du = -s \left(\text{erf}\left(\frac{\ln\left(-\frac{x^\tau-(d-m)s}{s}\right)}{\sqrt{\pi}}\right) - \text{erf}\left(\frac{\ln(d-m)}{\sqrt{\pi}}\right)\right).
\]

where \(\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_{0}^{x} e^{-t^2} dt\) is the Gauss error function.

Furthermore, \(R(\cdot)\) is a recursive function we introduced to ease the computation by
avoiding the double summation, defined as follows:

\[
R(\tau) = \begin{cases} 
(1 + R(\tau - 1))e^{-\beta(x^\tau - x^{\tau-1})} & \text{if } \tau > 1, \\
0 & \text{if } \tau = 1.
\end{cases}
\] (4.11)

Modeling all sequences. Thus far, one could model a single student-assignment pair via SSHP based on its historical observations by maximizing the log-likelihood function defined in Eq. 4.8. However, as mentioned in the previous section, we represent some of the parameters (\(\Theta\)) in a matrix format for all student-assignment pairs and assume similarity among them, i.e. a low rank structure on the matrix. Specifically, we denote the set of the vector parameters as \(\phi = \{c, p, b, v\}\), and the set of the matrix parameters as \(\Theta = (A, M, \Gamma^h, \Gamma^o, \Gamma^d)\) and impose a low-rank structure on all \(\Theta\) in our objective function. By using trace norm as a surrogate for low-rank structure, we constraint the trace-norm of \(A = (\alpha_{ij})_{U\times N}, M = (m_{ij})_{U\times N}, \Gamma^h = (\gamma^h_{ij})_{U\times N}, \Gamma^o = (\gamma^o_{ij})_{U\times N},\) and \(\Gamma^d = (\gamma^d_{ij})_{U\times N}\) to be small.

Loss for all sequences. Finally, we can formulate the objective function as follows, based on the collection of observed sequences \(O = \{X_{ij}, \text{s.t. } |X_{ij}| > 0\}\):

\[
\min_{\Theta, \phi} \mathcal{L} = -\frac{1}{|O|} \sum_{X_{ij} \in O} l(X_{ij}; \Theta_{ij}, \phi_i) 
\text{s.t. } A \geq 0, \Gamma_d \geq 0, \Gamma_o \geq 0, \Gamma_h \geq 0, c \geq 1, v > 0, 1 > b > 0
\]

\[
\text{tr}(\theta_u) \leq k_u, \text{ for } \theta_u \in \Theta_{ij}.
\] (4.12)

The main objective is the negative log-likelihood of observing all sequences with events, while the non-negative constraint on \(A\) is introduced to fit the definition of Hawkes that the sequences are self-exciting. All coefficients of the 3 types of external stimuli are also set to be non-negative. \(c\) is constrained to be greater than or equal to 1 to make sure the non-negative effect of student habit with the use of sinusoidal function, and each \(v\) element is the shape parameter in the reversed log-normal function thus needs to be positive. Each cell of \(b\) is set to be constrained between 0 and 1 to meet the assumption that the effect of assignment opening is decaying but not increasing or unchanged. We also constrain each parameter \(\theta_u\)’s trace norm in the matrix format to be small, which is equivalent to constraining the rank of
θ_u to be less than or equal to k_u.

4.2.3 Parameter Inference

We adopt Accelerated Gradient Method (AGM) [88] framework for the inference of parameters. The key subroutines of AGM in our model can be summarized as follows. For a matrix format parameter θ_u ∈ Θ, the objective is to compute the proximal operator:

\[
\theta_u^* = \arg\min_{\theta_u} \mathcal{M}_{\gamma,\theta_u^S}(\theta_u) = \arg\min_{\theta_u} \frac{\gamma}{2} \|\theta_u - P_{\theta_u}(\theta_u^S - \frac{1}{\gamma} \nabla_{\theta_u} \mathcal{L})\|_F^2.
\]

γ is the step size, θ_u^S is used to denote the current search point of θ_u, and \(\nabla_{\theta_u} \mathcal{L}\) is the gradient of loss \(\mathcal{L}\) w.r.t \(\theta_u\). \(P_{\theta_u}(\cdot)\) is a projection function to make sure the parameter value at each step is properly constrained. More specifically, \(P_{\theta_u}(\cdot)\) for all \(\theta_u \in \Theta\) is set to be \((\text{TraceProj}(\cdot))_+\) where the inner \(\text{TraceProj}(\cdot)\) is a trace projection [14] and the outer \((\cdot)_+\) projects negative values to 0. Similarly, the key subroutine for the inference of ϕ_u ∈ ϕ is shown as follows:

\[
\phi_u^* = \arg\min_{\phi_u} \mathcal{M}_{\phi_u^S,\gamma}(\phi_u) = \arg\min_{\phi_u} \frac{\gamma}{2} \|\phi_u - P_{\phi_u}(\phi_u^S - \frac{1}{\gamma} \nabla_{\phi_u} \mathcal{L})\|_F^2.
\]

\(P_{\phi_u}(\cdot)\) is also a projection function that makes sure the constraint of \(\phi_u\) is met. When a value falls out of the constrained interval, it is projected to the closet value within the interval.

4.3 Experimental Evaluation

To evaluate the performance of SSHP, in the following section, we present our preliminary experiments on RCHawkes-Gamma and several state-of-the-art point process models on synthetic and real-world datasets.
4.3.1 Datasets

Synthetic Data: Presuming 500 students and 20 assignments, we created $10^4$ ($500 \times 20$) simulated student-assignment pairs, and sampled $\sim 100$ events for each pair using the Ogata thinning algorithm [89], which is the most commonly used sampling method in the related literature. Specifically, we used the intensity function defined in Eq. 4.6 and sampled each of its parameters from normal distributions, where $A \sim \mathcal{N}(0.4, 0.1)$, $M \sim \mathcal{N}(0.5)$, $\Gamma^d \sim \mathcal{N}(15, 3)$, $\Gamma^o \sim \mathcal{N}(5, 3)$, $\Gamma^h \sim \mathcal{N}(0.5, 0.1)$, $v \sim \mathcal{N}(20, 10)$, $b \sim \mathcal{N}(0.5, 0.3)$, $p \sim \mathcal{N}(6, 4)$, and $c \sim \mathcal{N}(1.2, 0.1)$. We empirically set these distributions to approximate the intensity patterns observed in real data. For visualization, Fig. 4.1 shows a sequence generated by open library tick [6], in which all the parameters are set to be the means. The solid blue line shows the sequence intensity, where each blue dot represents a sampled activity, the dashed orange line is the base rate, and the synthetic deadline is 80, shown as the vertical red line. To check models’ sensitivity to the sparsity of the data, we created two sets of data by randomly masking 10% (named as Syn-10 data) and 90% (named as Syn-90 data) of the sequences to be unobserved. In other words, 10% of the sequences from Syn-10 dataset, and 90% of the sequences from Syn-90 dataset are unobserved.

Real Datasets: In this work, we use the same two real-world datasets as introduced in the previous section, namely CANVAS and MORF.
4.3.2 Baseline Approaches

We use all the state-of-the-art point process methods as baseline approaches, as introduced in Sec 3.3. Since the proposed model SSHP aims to model different external triggering effects, besides these baseline approaches, we also include the Homogeneous Poisson process [67] as a baseline because it can be viewed as the simplest category of point processes that captures external effects as constant. A summary of these baseline approaches is shown in Tbl. 4.1, from four aspects, namely if they model self-excitement, if they model non-homogeneous external triggers if they can deal with sequences with missing history, and if they have been used education domain.

<table>
<thead>
<tr>
<th>Model</th>
<th>Self-exciting</th>
<th>Non-constant base of time</th>
<th>Infer completely missing seq.</th>
<th>Application in Education</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poisson</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>HRPF</td>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>RMTPP</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>ERPP</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>DHPHR</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HPLR</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>EdMPH</td>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>SSHP</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

4.3.3 Experiment Setup

To test the model’s performance in predicting the future with and without historical observations respectively, we split our data into the following 3 sets: training set that contains the initial historical observations, which is used to train the model for parameter inference; partially missing test set that contains the rest of the historical observations, that is used for testing the first scenario. Finally, the completely missing test set contains the entire observations of the sequences, and it is used to examine models’ ability in generating personalized and accurate predictions for unobserved sequences, i.e. sequences without historical observations. For Syn-10, we naturally set the 10% masked sequences to be the completely missing test set. In the remaining 90% unmasked sequences, we use the first 70% of the activities (i.e. synthetic past observations) to be training and the later 30% (i.e. synthetic future activities to be predicted) to be partial missing testing. We
perform a similar procedure on Syn-90, with 90% masked sequences to be completely missing test set, and a 70% – 30% split in the remaining sequences for training and partially missing testing respectively. For both real-world datasets, we randomly holdout 20% of the sequences to be completely missing, and for the rest of the 80% sequences, we also use the same 70% – 30% split to generate training and partially missing testing. For the baseline models that are not able to generate personalized predictions of future times without historical observations (i.e. Poisson, RMTPP, ERPP, and EdMPH), we report the root mean squared error (RMSE) of the time prediction on the partially missing test set only, and for the other models, we report the RMSE on both partially and completely missing test sets. The hyperparameters of the proposed SSHP across all datasets are tuned via grid search, with $\beta \in \{0.01, 0.05, 0.1, 0.5, 1, 5, 10\}$ and $S \in \{\frac{1}{5}, \frac{1}{3}, 1, 2, 3, 5, 7\}$.

4.3.4 Experimental Evaluation I: Parameter estimation

As a way to evaluate SSHP’s performance in capturing the sequence dynamics, we investigate its ability to find the true parameters of the underlying processes. Since these parameters are available from the synthetic datasets, we calculate the root mean squared error (RMSE) between the estimated parameter values by SSHP and the actual parameter values that have been used to generate the synthetic datasets. The results are shown in Tbl. 4.2. Generally, SSHP performs better in the partially missing test set than in the completely missing test. That is because the task of learning completely unobserved sequences without histories is more challenging than learning sequences with partially observed histories. Additionally, the results show that the RMSEs in Syn-90 dataset are only marginally higher than in Syn-10 in both partially and completely missing test sets. This suggests the model’s robustness and its potential to recover the parameters even when the ratio of unobserved sequences is high in the dataset.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>v</th>
<th>b</th>
<th>p</th>
<th>c</th>
<th>A</th>
<th>M</th>
<th>$\Gamma^d$</th>
<th>$\Gamma^o$</th>
<th>$\Gamma^h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syn-10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>part. miss.</td>
<td>1.33</td>
<td>0.1</td>
<td>1.33</td>
<td>0.09</td>
<td>0.05</td>
<td>2.64</td>
<td>1.65</td>
<td>1.08</td>
<td>0.16</td>
</tr>
<tr>
<td>compl. miss.</td>
<td>1.23</td>
<td>0.12</td>
<td>1.39</td>
<td>0.16</td>
<td>0.13</td>
<td>2.60</td>
<td>2</td>
<td>1.54</td>
<td>0.13</td>
</tr>
<tr>
<td>Syn-90</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>part. miss.</td>
<td>1.34</td>
<td>0.10</td>
<td>1.33</td>
<td>0.09</td>
<td>0.06</td>
<td>2.39</td>
<td>1.80</td>
<td>1.14</td>
<td>0.18</td>
</tr>
<tr>
<td>compl. miss.</td>
<td>1.31</td>
<td>0.12</td>
<td>1.38</td>
<td>0.16</td>
<td>0.12</td>
<td>2.61</td>
<td>1.97</td>
<td>1.51</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Table 4.2: RMSE of parameters learned by proposed model SSHP in synthetic datasets.
To provide a visual representation of these results, Fig. 4.2 shows the sampled intensity of a real sequence in Syn-90 dataset and the predicted intensity that is sampled based on the predicted parameters. This figure demonstrates the model’s ability to accurately capture the dynamics of the sequence.

![Figure 4.2: Predicted Intensity of a synthetic sequence.](image)

### 4.3.5 Experimental Evaluation II: Time Prediction

We evaluate the model performances in predicting the next 10 future activities after the observation window is ended, using RMSE between the actual and predicted times as our measure. As the number of future activities grows, the task of predicting their arrival times becomes more challenging.

We first present the experiment results for SSHP and baseline approaches on synthetic datasets. Fig. 4.3 shows the model performances in the partially missing test set in both Syn-10 and Syn-90, while Fig. 4.4 shows the performances in the completely missing test set. The x-axis represents the future events’ indices. For example, \(x = 2\) represents the second event in the future after the end of the observation period \(T\). The y-axis is RMSE of time predictions in the log-scale, for a clearer separation between the models in the figures. Some baselines, such as ERPP and RMTPP, are missing from the lower plots since they cannot predict unobserved sequences (student-assignment sequences in the completely missing test set). We can see that SSHP clearly achieves the smallest RMSE of time predictions compared to the baseline approaches in all settings. Even though neural models
ERPP and RMTPP start to show better performances in later event predictions, they are not able to predict unobserved sequences (i.e. completely missing test set). As expected, since recovering completely unobserved sequences is more challenging, SSHP’s performance on the partially missing test set is better than its performance in the completely missing test set.

Next, we evaluate each model’s performance using the two real-world datasets. It is
worth mentioning that the observed history in MORF is the shortest among all datasets, having an average of less than $\sim 26$ observations per sequence, and $\sim 18$ observations for training. For this reason, the prediction window is set to be 8 in MORF instead of 10 to achieve meaningful evaluation. The evaluation in partially missing test set and completely missing test set is respectively presented in Fig. 4.5 and Fig. 4.6. As it is shown in the figures the proposed SSHP model outperforms the baseline approaches, especially by a big margin in Canvas’s completely missing test set. This is consistent with the synthetic dataset results. In contrast, the performances of neural models ERPP and RMTPP are not as promising as they are in synthetic datasets, especially in MORF. One possible explanation is that the short training sequences in MORF restrict the ability of neural-based models.

Another observation is that for higher indexed events in MORF’s completely missing set and for lower indexed events in Canvas’s partially missing test set, we observe overlapped confidence intervals between HPLR and SSHP, suggesting a less significant difference between the two models’ performances. However, as the large confidence interval in HPLR suggests, its results are not robust and vary too much in the experiments. A potential explanation for the good predictions in HPLR is that for some student-assignment pairs the activity dynamics are rather invariant and a constant base rate, as in HPLR, is sufficient to capture them.

![Figure 4.5: Time prediction RMSE on partially missing test set with 95% confidence interval on two real-world datasets.](image)

In conclusion, SSHP has shown to have superior time prediction performance in both
Figure 4.6: Time prediction RMSE on completely missing test set with 95% confidence interval on two real-world datasets.

synthetic and real-world datasets comparing with baseline approaches, especially on the challenging task of predicting the future for the completely missing test set.

4.3.6 Experimental Evaluation III: Ablation Study on the Importance External Stimuli

To verify each component’s importance in the intensity function, we compare SSHP to its variations SSHP-s, SSHP-o, SSHP-h, and SSHP-d, which respectively represents the model achieved by taking out the following components: self-excitement $s(t)$, effect of assignment opening $\mu_o(t)$, effect of student habit $\mu_h(t)$ and effect of deadline $\mu_d(t)$.

Fig. 4.7 and Fig. 4.8 show the performance of these models in comparison with each other and with SSHP in respectively partially and completely missing test set. In general, SSHP achieves lower time prediction errors in both real-world datasets, indicating the importance of each individual component. Furthermore, in the partially missing test set as shown in Fig. 4.7, while the improvement of modeling self-excitement is only marginal comparing with SSHP in CANVAS (left figure), self-excitement is shown to be a major factor in MORF (right figure), as SSHP-s has higher prediction errors in MORF than other variations. Additionally, as shown in Fig. 4.8, we can see the differences between SSHP and its variations are much more distinct in the completely missing test set (i.e. when the history is unobserved).
Figure 4.7: Time prediction RMSE with 95% confidence interval of SSHP and variations on partially missing test set.

Figure 4.8: Time prediction RMSE with 95% confidence interval of SSHP and variations on completely missing test set.

More specifically, in CANVAS dataset (left figure), we see that SHPP-d’s error is the highest among all models. This shows strong evidence of the deadlines’ effect on student activities in CANVAS, which also suggests the importance of modeling $\mu_d(t)$. On the other hand in MORF, when comparing SSHP and SSHP-h (right figure), we can see that the effect of student habit is not presented at the beginning of the sequence, as the error is lower when this stimulus is not included. However, the importance of including student habits in the model is significant after the second event. Another interesting observation is the higher confidence
interval presented in SSHP-o. One explanation is that some students are more sensitive to assignment opening compared to the others, therefore excluding $\mu^o(t)$ from the equation can cause a higher error to some sequences but not the others. The difference that is observed in the components’ importance in the two datasets can come from the different nature of the two educational systems and the presented courses. For example, one expects the effect of the deadline to be more prevalent in courses with a high late-submission penalty, compared to the ones with a more flexible scheme. To conclude, despite the different characteristics that have been unveiled in the two datasets, we can see that all three external stimuli and the self-excitement components are important in modeling student activities.

4.4 Discovering Procrastination Patterns via SSHP

4.4.1 Clusters of Learning Dynamics Captured by SSHP

First, we investigate if the learned parameters can describe students’ behaviors in assignments in a meaningful way that shows their cramming and procrastination behaviors. To do so, we cluster all student-assignment pairs via K-Means clustering algorithm, representing each of student-assignment item as its learned parameters: $(u_i, a_j) = (\alpha_{ij}, m_{ij}, \gamma^d_{ij}, \gamma^o_{ij}, \gamma^h_{ij}, v_i, b_i, p_i, c_i)$.

To find the optimal number of clusters, we use the elbow method on clustering loss. In both CANVAS and MORF datasets, the achieved optimal cluster number is 3, which means that 3 student-assignment interaction patterns are uncovered in both datasets. Figures 4.9 and 4.10 show the parameter values for cluster centers in CANVAS and MORF datasets, respectively. For a clearer presentation, $m$ is scaled down by 24 (time unit changes from hours to days) and $\alpha$ and $b$ are scaled up by 10 in the figures respectively. Error bars show the 95% confidence interval within each cluster.

Specifically, by comparing CANVAS 3 (cluster 3 in CANVAS dataset) with clusters 1 and 2 in CANVAS, we can see that the interactions between students and assignments in CANVAS 3 are shown to be less sensitive to the deadline until much later when it is too close to the deadline (smaller $v$ and larger $\gamma^d$). Also, negative $m$ in CANVAS 3 indicates late submissions or other assignment-related activities, after the deadline. Not only that but the burstiness of the events in this cluster is also shown to be higher than other clusters (larger $\alpha$). One possible explanation is that the students in this cluster procrastinated on the
Figure 4.9: Clusters of student learning dynamics characterized by SSHP in CANVAS.

Assignments in it and only started to work on them much later than they should have, which explains the bursty and intense activities close to the deadline. Furthermore, we can see that the effect of assignment opening or availability wears off much faster in CANVAS 3 (smaller $b$), meaning that the period of time that this cluster is affected by assignment opening is shorter. This suggests that overall, this cluster is less sensitive to the assignment opening.

When it comes to student habit, we see that the peak of periodicity shows up at a later time (large $p$), indicating that the students in CANVAS 3 interact with the course usually later during the day, comparing with CANVAS 2 and 1. On the other hand, even though the differences are shown to be smaller when comparing CANVAS 1 and CANVAS 2 clusters, many of them are significant. Particularly, the results clearly show that CANVAS 2 is more sensitive to the deadline in the sense that assignment-related activities are finished much earlier (larger positive $m$ and $\gamma^d$). Their base activities triggered by student habits are also shown to be more intense (higher $c$) even though their peak time is usually later during the day (larger $p$). So to conclude, the learning pattern in CANVAS 3 suggests procrastinating-like behaviors, with less sensitivity to the deadline and the assignment opening, as well as more bursty and intense behaviors. On the other hand, learning patterns in CANVAS 2
suggests an “early birds” type of learning behavior, in which assignment-related activities are finished earlier by around 4 days. Also, they tend to be more sensitive to the opening of assignments, with less bursty behaviors, which can be interpreted as an opposite behavior of procrastination.

Similarly, in MORF (Figure 4.10), different characteristics are uncovered by the discovered clusters. We can see that the effect of the deadline starts late and ends late (smaller $v$ and smaller negative $m$) for MORF 1 and 3. Also, student activities on assignments in MORF 1 and 3 are more bursty (larger $\alpha$). On the other hand, MORF 2 is more sensitive to the effect of assignment opening for a longer period of time (larger $b$), and student habits also seem to have a stronger effect on MORF 2, suggested by larger $c$ and $\gamma^h$. Overall, we can conclude that MORF 2 activity patterns represent the “early birds” type and MORF 1 activity patterns show the most procrastination-like behaviors among the 3 clusters.

By comparing the clusters where procrastination-like behaviors are suggested between the two datasets, we can see that the parameters show different strategies in them. Specifically, in MORF 1, less bursty and more delayed submissions are observed (smaller negative
Figure 4.11: Clusters of student learning dynamics characterized by SSHP in CANVAS and MORF.

$m$ and smaller $\alpha$) than in CANVAS 3, which can be an indication of procrastination. Another potential explanation for this difference can be the different nature of the courses as we mentioned in the section of ablation study, where the penalty of late submissions can be stronger in CANVAS than in MORF.

4.4.2 Association between Procrastination Behaviors and Grades

To show the association between student activity patterns on assignments and their performance in them, we check the student grades on assignments in each cluster. The results are presented as box plots shown in Fig. 4.11. As we can see, median grades in CANVAS 3 and MORF 1 are visibly smaller than other clusters in their datasets. But also, the distribution of grades in each two clusters is different. To see if the differences of grade distributions between clusters are significant, for each of the datasets, we conduct a Kruskal-Wallis test on the grades between any two clusters discovered by SSHP. We find out that all the p-values are significantly smaller than 0.05, suggesting significant differences in the grade distribution between all clusters. Combining these observations with the conclusions from Figures 4.9 and 4.10, we see that clusters that show procrastination behaviors with less sensitivity to the deadlines and assignment openings (CANVAS 3 and MORF 1) also are shown to have significantly lower grades. We can conclude that clusters with more procrastination-like beh-
behaviors are associated with lower grades in both datasets. This demonstrates that SSHP can capture underlying student activity patterns with meaningful parameters that can be used as good indicators of procrastination behaviors and student performances.

4.5 Applying SSHP to Recommendation Datasets

To this end, our experiments and analyses demonstrated the great fit of our model in student procrastination modeling. In this section, we extend and apply our model to the domain of recommendation. In particular, we consider the Foursquare user check-ins in NYC and Tokyo datasets, i.e. Foursquare-NYC and Foursquare-TKY described and used in Ch. 3.3.

As the stimuli deadline and assignment opening are not available in these datasets, we adjust our model SSHP to only consider user habitual behaviors by setting the coefficients $\Gamma_d$ and $\Gamma_o$ of stimuli deadline and assignment opening as 0. Consider the possibility that different seasonal checking-in behaviors of users may exist towards different types of venues, we relax our original assumption that each student has the same habits towards all assignments, and allow our model to learn $P$ and $C$, which are respectively the matrix formats of $p$ and $c$ that represented peaks and the minimum number of activities. More specifically, the objective of this formulation is given as below:

$$\min_{\Theta, \phi} \mathcal{L} = -\frac{1}{|O|} \sum_{X_{ij} \in O} l(X_{ij}; \Theta_{ij}) \quad (4.15)$$

s.t. $A \geq 0, \Gamma_h \geq 0, C \geq 1$,

$$tr(\theta_u) \leq k_u, \text{ for } \theta_u \in \{A, \Gamma_h, C, P\},$$

where

$$l(X; \theta) = \log L(\theta) = \sum_{\tau=1}^{K} \log(\lambda(x^\tau)) - \int_{0}^{x^K} \lambda(u)du$$

$$= \sum_{\tau} \log \left(\gamma^h \mu^h(x^\tau) + \alpha \beta R(\tau) \right) - \gamma^h U^h(x^K)$$

$$+ \alpha \sum_{\tau=1}^{K} (e^{-\beta(x^K-x^\tau)} - 1).$$

62
First, to see if our model fits to the new domain application, we conduct a preliminary study which exams the goodness-of-fit of temporal point process, based on the random time change theorem [38], which states the following:

**Theorem 4.5.1.** Say $t_1, t_2, ..., t_K$ is a realisation over time $[0, T]$ from a point process with conditional intensity function $\lambda$. If $\lambda$ is positive over $[0, T]$ and $\Lambda(T) < \infty$ a.s. then the transformed points $\{\Lambda(t_1), \Lambda(t_1), ..., \Lambda(t_K)\}$ form a Poisson process with unit rate, where $\Lambda(t) = \int_0^t \lambda(s)ds$.

This test is equivalent to test Poissonianess of $\{\Lambda(t_1), \Lambda(t_1), ..., \Lambda(t_K)\}$, which can be realized by running a hypothesis test to check $\tau_i$ is drawn from $\text{Exp}(1)$, where $\tau_i = \Lambda(t_i) - \Lambda(t_{i-1})$. To see this, we ran Kolmogorove-Smirnov test and obtain the mean and the standard deviation of P-values to be 0.06 and 0.14 in Foursquare-TKY, and 0.034 and 0.095 in Foursquare-NYC. These result suggest we can not accept the hypothesis that $\{\Lambda(t_1), \Lambda(t_1), ..., \Lambda(t_K)\}$ forms a Poisson process with unit rate, and consequently shows that the model does not fit the data well according to Theorem 4.5.1.

This observation suggests that the modeling describe in this section does not well-describe the underlying distribution of user-venue check-in sequences. A possible explanation is that a shifted sine function used in this model is too strict for the modeling of the external effects thus restricts model’s power of learning self-excitement and external effect accurately. Therefore, a different external-stimuli-modeling framework may be beneficial in the future for the task of user check-ins modeling in the domain of recommendation.

### 4.6 Conclusion

To answer RQ2, we proposed a novel stimuli-sensitive Hawks process model (SSHP) to represent student’s cramming and procrastination behaviors in online courses, according to their activities. Our model captures three types of external stimuli in addition to the internal stimuli between activities, i.e., the effect of assignment deadline, assignment availability, and student’s personal habits. SSHP models all student-assignment pairs jointly, which enables the model to generate personalized predictions for both partially missing and completely missing activity sequences. Our experiments on both synthetic and real-world datasets demonstrated SSHP’s superior performance comparing to the state-of-the-art base-
line approaches, especially in the more challenging task of future time prediction for time sequences where the history is completely missing. Our ablation studies on SSHP showed that each component of our model is necessary for achieving its superior performance. Finally, we demonstrated that not only SSHP excels at future time predictions, but also its model parameterization provides meaningful interpretations and insights into the association between students’ procrastination patterns and their grades. Particularly, we discovered 3 clusters of behaviors on assignments: one with stronger procrastinating behaviors, with less sensitivity to the deadline and the assignment opening, as well as more bursty and intense behaviors; another one with “early birds” type of learning behaviors, with more sensitivity to deadlines and less bursty behaviors; and a third one in between the two. We showed that grade distributions in these clusters have meaningful differences, with the lowest grades associated with procrastinating-like behaviors. On the other hand, considering the application of recommendation, our extended model is limited in its power in representing the external effects in user check-in datasets, and a different framework may be needed in the future for this task.
CHAPTER 5
Modeling Marked Temporal Point Processes

5.1 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 users’ interactive behavior, such as when do they have these activities (i.e. activity timestamps), what kind of activities they have (i.e. activity types), and how do the activities turn out (i.e. markers such as the ratings of the activities). For example, in an e-commerce platform, a user may purchase an item (activity) on Friday night (time) then rate it a five-star (marker), or in an online learning system, a student may attempt a quiz problem (activity) on Monday morning (time) and provide a correct answer (marker). The timings of these events, particularly the historical activity intensities and inter-arrival times can reveal meaningful trends in such data and be an essential factor in better representing users’ true interactive behaviors and predicting their upcoming activities.

Marked Temporal Point Process models (MTPPs) [59] are special types of Temporal Point Process models (TPPs) [28] that represent event timings along with their associated markers. Due to their ability to model events in continuous time, MTPPs are beneficial for application domains such as social networks, recommender systems, or education where modeling and predicting user activity timings are essential. 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.

Despite the successful application of MTPPs in various domains, they still suffer from two major limitations: ineffectiveness of capturing the influence of event dynamics on marker distributions and the lack of fine-grained representation of historical marker distributions.

First, most MTPPs can not efficiently model the interrelationships between marker distribution and activity dynamics. Traditional MTPPs usually empirically specify the con-
ditional 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 to fully capture the underlying dynamics of the data. Additionally, in
these methods, the strict parameterization of intensity functions risks model misspecification.
More recently, Recurrent MTPPs have been proposed to resolve the strict parameterization
issue in traditional MTPPs by using the RNNS’ hidden states in defining the intensity func-
tions. However, in the few recent attempts on MTPPs, the markers and activity timestamps
are either modeled with the same dynamics via a shared hidden state in one RNN [32] or as
two independent dimensions with no relations assumed between them at each step [23,129].
The first approach assumes that the marker sequence and the event timings follow 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 event 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 inade-
quate for such applications. For example, the increases of student learning performance in
an online course do not necessarily come with intense learning dynamics, and a less frequent
movie watching behavior does not always suggest the user does not like the movies they
watched. In other words, activity dynamics and markers do not necessarily follow the same
dynamics, and modeling them via the same representation may be inadequate for such more
complicated user cases (i.e. the first approach). Furthermore, a student’s future choices 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), or a user’s future choices
and preferences of movies may have been built up based on their movie watching experience
from the past. In other words, the interrelationship between markers and activity dynam-
ics over time may exist and not modeling this relationship may reduce model’s power in
understanding true user behavior (i.e. the second approach).

As the second limitation, modern neural-based MTPPs represent activity histories as
abstract state vectors in RNNS. This coarse-grained representation of history can result
in losing fine-grained activity-level features in RNN updates, and dilute 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 important task. The interpretability problem will be further complicated when inter-tangled with the interrelationships problem mentioned above. In that case, important interpretation of how marker and activity dynamics interrelate will be compromised, thus makes it even harder to explain the data dynamics.

To address the above limitations, we propose a novel neural MTPP model, Marked Poisso Processes via Memory-Enhanced Neural Networks (MoMENt), that captures and represents the bidirectional relations between markers and events, while revealing the detailed patterns in their dynamics. We propose recurrent activity updater (RAU) and Memory-Enhanced marker updater (MEMU) in MoMENt that respectively model activity dynamics and markers, and capture the marker influence on activities and vice versa (first limitation). MEMU also utilizes external memories via Memory-Augmented Neural Network (MANN) [47], to obtain a fine-grained representation of marker distributions (second limitation).

We formulate MoMENt based on the problem of user activity modeling in continuous time. Our extensive experiments on six real-world datasets from two application domains show consistent improvements of MoMENt over the state-of-the-arts, demonstrating the effectiveness of MoMENt in capturing user-system interacting dynamics and representing fine-grained user-system relation.

5.2 Problem Formulation

We focus on the problem of user activity modeling considering its importance and its wide range of applications, such as recommendation [71, 119, 120] and student learning modeling [23, 135, 137]. Suppose we are given \( N \) users and their interactions with \( Q \) items. The collection of all users’ interactive activities can be represented as \( S = \{S^1, ..., S^N\} \), with \( S^i \) representing user \( i \)'s sequence of activities. Also suppose that the activity at \( j \)-th step is associated with its timestamp \( t^j_i \), its type \( y^j_i \), item-level attributes, such as item id \( qid^j_i \) or item tag \( qf^j_i \), as well as the observed user-item interaction outcome \( r^j_i \), 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, qid_j, qf_j, r_j) | j = 1, ..., K\} \textsuperscript{5}. Note that some of these information may be missing for some activities, such as missing \(qf\), or unobserved \(r\), which will be padded as \(-1\). Our goal is to predict users’ future activity times, types, and outcomes, given their interaction history with the system 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 the future activities (i.e., \textit{activity2activity influence}), which corresponds to the time dependency assumption that has been used in many point process families, such as in Hawkes processes. For example, a user’s historical activities can trigger follow-up activities. Consider an example of a student learning from an online course. Watching a lecture video may lead to follow-up activities such as trying out related quiz questions. Second, we assume the influence of past markers on the future activities (i.e., \textit{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 user who has had a satisfactory purchase from a website in the past may interact with it more often in the future. Similarly, a student who has failed a quiz may follow up with activities such as reviewing related lectures.

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., \textit{marker2marker influence}). In other words, the marker distribution can change over time, as a reflection of evolving user-system relationship and the markers’ historical distributions. For example, user satisfaction, as a reflection of the user-system relationship can be evolving according to the user’s historical satisfaction with the system. Or, students’ past grades, as a reflection of their evolving knowledge of course concepts, can be predictive of their future grades. Finally, we assume the influence of historical activities on the markers’ future distribution (i.e., \textit{activity2marker influence}). For example, a series of intense browsing activities in the system may end up with the user’s satisfactory purchase. Likewise, a student’s historical learning activities can affect their future grades. The first two assumptions above have been widely used in the literature on point process modeling. However, the last two assumptions have been largely overlooked.

\textsuperscript{5}We omit user index \(i\) for presentation simplicity.
5.3 MoMENt

In this section, we formally introduce Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt). An overview of MoMENt is presented in Fig. 5.1.

Figure 5.1: 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.

At step $j$, in the input layer, the 5-tuple $(t_j, y_j, qid_j, qf_j, r_j)$ is used as input and embedded to obtain activity dynamic embedding $e_{a_j}$ and marker embedding $e_{m_j}$ (Ch. 5.3.1). 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) (Ch. 5.3.3). In parallel, Memory-Enhanced Marker Updater (MEMU) is proposed to model the influences of historical markers and activity dynamics on the future marker dis-
tributions, i.e., marker2marker (dash-dotted red arrow) and activity2marker (dotted purple arrow). Inspired by Memory-Augmented Neural Networks (MANNs), MEMU utilizes external memory matrices to represent each user’s evolving relationship with all the items, to provide a fine-grained explanation of marker distribution (Ch. 5.3.2).

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 (Ch. 5.3.4). Each model component’s details are provided below.

5.3.1 Input Layer

For activity type $y_j$, we first represent it as a one-hot vector $y_j$. As point processes can have many types, we optionally apply a linear transformation on $y_j$ to obtain a more compact activity type representation $e_y = W_y y_j$. Furthermore, we represent the timing of activity $x_j$ as the inter-arrival time between $j$-th activity and $(j-1)$-th activity. 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 to the definition of point processes which are defined based on inter-arrival times. Finally, we use $e_a = [x_j; e_y]$ to represent activity dynamic embedding. To encode item id $qid_j$, we generate its one-hot representation $qid_j$. For categorical item feature $qf_j$, we also generate its one-hot representation $qf_j$ if it is singular, otherwise we apply Multi Label Binarizer. For categorical marker $r_j$, we similarly first obtain its one-hot representation $r_j$. As each user-item interaction outcome depends on the item, we concatenate item-level features with marker to obtain the marker embedding $e_r = [qid_j, qf_j, r_j]$. To also encode time information and to provide more interpretability, similar to [75], we obtain our final time-masked marker embedding via $e_m = e_r \circ \sigma(W_m \phi(x) + b_m)$, where the second term is a time mask computed as a function of time representation $x$ after a sigmoid function. This mask is element-wise product with marker representation $e_r$, so that less important historical activities and their influence will be masked.
5.3.2 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 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 relates to user-system relationship, and also is hard to interpret. Memory-Augmented Neural Networks (MANNs), on the other hand, are a form of neural networks that have shown to achieve more effective performances than standard RNNs while providing fine-grained and meaningful interpretations [47]. MANNs achieve this by using external memory matrices to enable read and write operations [47, 107] that provide structured memory slots leading to local state storage and transactions. However, MANNs have not been considered for MTTPs.

Motivated by the limitation of using RNNs to model marked point processes, we propose to use Key-Value Memory Network [83] 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 \( \mathbf{M}^v_j \in \mathbb{R}^{C \times d_v} \) to model user-item relationship with all \( C \) item components at step \( j \), described by \( d_v \) latent variables. \( \mathbf{M}^v_j \) is set to be dynamic to capture 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 value to key (e.g., students’ knowledge of all concepts as an explanation of learning outcomes).

We define MEMU as \( \mathbf{M}^v_j = \text{MEMU}(\mathbf{e}_{mj}, \mathbf{h}_{j-1}, \mathbf{M}^k, \mathbf{M}^v_{j-1}) \), where the current state of marker \( \mathbf{M}^v_j \) is updated by the current marker embedding \( \mathbf{e}_{mj} \), the previous hidden state of activity dynamics \( \mathbf{h}_{j-1} \) (will be introduced in the section of RAU), static key matrix, and previous state of marker \( \mathbf{M}^v_{j-1} \). An illustration of MEMU is given in Fig. 5.2.
Specifically, MEMU is defined as follows:

\[ e_j = [h_{j-1}; e_{mj}], \quad (5.1) \]
\[ q_j = W_q e_j, \quad (5.2) \]
\[ a_j = \text{softmax}(\frac{M^k q_j}{\sqrt{d_c}}), \quad (5.3) \]
\[ V_j = W_V e_j u_v^\top, \quad (5.4) \]
\[ s_j = (a_j^\top V_j)^\top, \quad (5.5) \]
\[ M^v_{vj} = \tanh(W_M M^v_{j-1} + s_j u_M^\top + b_M), \quad (5.6) \]
\[ Z_j = \sigma(W_Z M^v_j + s_j u_Z^\top + b_z), \quad (5.7) \]
\[ M^v_j = Z_j \odot M^v_{j-1} + (1 - Z_j) \odot \tilde{M}^v_j. \quad (5.8) \]

In Eq. 5.1, we first concatenate the current marker embedding \( e_{mj} \) with previous hidden state \( h_{j-1} \) from RAU to obtain \( e_j \). The resulted embedding is multiplied with embedding matrix \( W_q \) to obtain an embedding vector \( q_j \) of length \( d_k \), via Eq. 5.2. A scaled dot-product \cite{115} is applied to the resulted embedding \( q_j \) and key matrix \( M^k \), followed by a softmax via Eq. 5.3, to obtain a weight vector \( a_j \) of length \( C \). The obtained weight \( a_j \) can be interpreted as the correlation between current item and all item components, considering
user’s historical interactions. Next, by using Eq. 5.4, the original embedding vector \( \mathbf{e}_j \) is embedded to 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 Eq. 5.5 to obtain the weighted fine-grained outcome in terms of \( C \) components, considering historical influences of interactions.

The resulting representation of outcome is then used to update user-item relationship in the next state, i.e. \( \mathbf{M}^v_{j+1} \), via Eq. 5.6-5.8 following a routine similar to GRU [24], to allow “adding” and “erasing” the historical effects of both user interaction dynamics (i.e. \( \mathbf{s}_j \) computed based on \( \mathbf{h} \)) and relationship with the items (i.e. \( \mathbf{M}^v_j \)).

It is worth mentioning that the aforementioned structure is particularly designed to achieve the model capacity that TPPs can generally offer, namely modeling the historical influence of each of the previous activity (Eq. 5.1-5.3) considering possible complicated data dynamic (Eq. 5.4-5.8). While on top of that, it also provides extra interpretations that conventional TPPs often overlooked, namely fine-grained marker2marker and activity2marker influences via MANN structure.

### 5.3.3 Recurrent Activity Updater (RAU)

Now we introduce the framework of our Recurrent Activity Updater (RAU), defined as \((\mathbf{h}_j, \mathbf{c}_j) = \text{RAU}(\mathbf{e}_{aj}, \mathbf{h}_{j-1}, \mathbf{c}_{j-1}, \mathbf{M}^v_j)\). 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 are in Long Short Term Memory (LSTM) [54]. The input data on the other hand contains the current activity embedding \( \mathbf{e}_{aj} \), 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 \( \mathbf{M}^v_j \). The core idea of this updater is to use the framework of LSTM to update \( \mathbf{h}_j \) taking into account the effect of historical activities in terms of their timings and types to the future activities (i.e. \( \mathbf{h}_{j-1} \) and \( \mathbf{c}_{j-1} \)) as well as the historical influence of markers on the future activities (i.e. \( \mathbf{M}^v_{j-1} \)). An illustration of RAU
is given in Fig. 5.3. More specifically, RAU is defined via the following equations:

\[ i_j = \sigma(W_i e_{aj} + U_i h_{j-1} + V_i c_{j-1} + P_i M_j^v y_1 + b_i), \]  
\[ f_j = \sigma(W_f e_{aj} + U_f h_{j-1} + V_f c_{j-1} + P_f M_j^v y_f + b_f), \]  
\[ c_j = f_j \odot c_{j-1} + i_j \odot \tanh(W_c e_{aj} + U_c h_{j-1} + V_c c_{j-1} + P_c M_j^v y_c + b_c), \]  
\[ o_j = \sigma(W_o e_{aj} + U_o h_{j-1} + V_o c_{j-1} + P_o M_j^v y_o + b_o), \]  
\[ h_j = o_j \odot \tanh(c_j). \]

As we can see in the above equations, similar to LSTM, considering the previous influence of activities (i.e. \( h_{j-1} \)) and markers (i.e. \( M_j^v \)), RAU updates its input state \( i \), forget state \( f \), cell state \( c \), output \( o \), and hidden state \( h \) respectively in Eq. 5.9-5.13. We see that during the computation of 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 long term dependencies of historical activities and markers.

\[ \text{Figure 5.3: An illustration of RAU.} \]
5.3.4 Prediction Layer

5.3.4.1 Activity Time Prediction.

In this section, we aim to predict when a user will have 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 $\lambda^*$ of time conditioning on the history of activities. Naturally, since $h_j$ represents the historical influence of activity dynamics, the intensity function of user interaction sequence can be represented as a function of $h_j$. More specifically, we propose to use the following function to define $\lambda^*$:

$$
\lambda^*(t) = \text{relu}\{b_b + \alpha(h_j) \exp \left( - \beta(h_j)(t - x_j) \right) \}, \text{ where (5.14)}
$$

$$
\beta(h_j) = (w_\beta^\top h_j)_+, \text{ (5.15)}
$$

$$
\alpha(h_j) = \tanh(w_\alpha^\top h_j). \text{ (5.16)}
$$

In Eq. 5.14, $b_b$ can be interpreted as the base rate, or the number of activities that naturally arrive due to external effects; $\alpha(h_j)$ defined in Eq. 5.16 describes the number of activities at $t$ that arrive due to the influenced by the previous activities (e.g. past activities triggered future ones). For activation function, we choose tanh due to its flexibility in transforming the values to the range of $[-\infty, +\infty]$, to capture possible time dependencies such as self-triggering (i.e. $\alpha > 0$), self-inhibiting (i.e. $\alpha < 0$), or memoryless (i.e. $\alpha = 0$) for each time step. Finally, $\beta$, as defined in Eq. 5.15, explains how fast the influence of the past activities decay over time, described by an exponential function in Eq. 5.14. To model this decay rate, we first linearly transform $h$ by multiplying it with a weight vector $W_\beta$, and then project it to the positive space, denoted as $(\cdot)_+$, assuming that the past influence is decreasing 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 process that restrict $\alpha$ to be positive (i.e. strictly triggering effects) and invariant to time, $\alpha(h_j)$ defined above allows the internal influence from past activities to be negative, or 0, which provides a more flexible representation of complex time dependencies such as self-inhibiting or Poisson-like.

2. Comparing with more recent Hawkes process models with neural representations that are hard to interpret, this intensity function has the interpretation of several key phenomena described in traditional point processes,
such as the decaying internal influence that are characterized by $\alpha$ and $\beta$.

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

$$ f^*(t) = \lambda^*(t) \exp \left( - \int_{t_j}^{t} \lambda^*(\tau) d\tau \right), \quad (5.17) $$

$$ g_{j+1} = \tanh(w_g^\top \log(f^*(t_j)) + b_g), \quad (5.18) $$

$$ \hat{x}_{j+1} = \text{relu}(w_t^\top g_{j+1} + b_t), \quad (5.19) $$

$$ \hat{t}_{j+1} = t_j + \hat{x}_{j+1}. \quad (5.20) $$

More specifically, we first obtain the *conditional arrival distribution* $f^*(t)$ that describes the distribution of inter-arrival times via Eq. 5.17 that describes its relationship to intensity function $\lambda^*$. Once we have $f^*(t)$, we can predict inter-arrival time $\hat{x}_{j+1}$ using Eq. 5.18-5.19. We feed the log density $\log(f^*(t_j))$ to a fully connected network. We choose relu as the last activation function to guarantee the idle time to be non-negative. The prediction of next activity time can be then computed via Eq. 5.20 which sums up current activity time $t_j$ and the predicted next inter-arrival time $\hat{x}_{j+1}$.

### 5.3.4.2 Activity type prediction.

To predict the type of $(j+1)$-th activity $\hat{y}_{j+1}$, we compute the probability of $y_{j+1}$ being type $y_d \in \mathcal{Y}$ via softmax function defined as below:

$$ P(y_{j+1} = y_d | h_j) = \frac{\exp(w_y^\top h_j + b_y)}{\sum_{y_d \in \mathcal{Y}} \exp(w_y^\top h_j + b_y)}. \quad (5.21) $$

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

### 5.3.4.3 Marker Prediction

For the prediction of marker $r_{j+1}$, namely user’s interaction outcome at step $j + 1$, we first obtain user’s current relationship with systems via Eq. 5.22 to get a column vector $v$, which is then concatenated with item-level features and passed through a fully-connected network with activation function $\phi_1$ of choice so that both item information and user’s
relationship to the items can be encoded in Eq. 5.23. And the final prediction of marker $\hat{r}_{j+1}$ is computed via Eq. 5.24 using another fully connected layer, where $\phi_2$ can be activation functions such as sigmoid for binary markers.

\[ v_j = (a_j^\top \mathbf{M}_j^i)^\top, \quad (5.22) \]

\[ n_{j+1} = \phi_1(w_r^\top [v_j; \mathbf{q}d_{j+1}; \mathbf{q}f_{j+1}]), \quad (5.23) \]

\[ \hat{r}_{j+1} = \phi_2(w_p^\top n_{j+1} + b_p). \quad (5.24) \]

### 5.3.5 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:

\[ \mathcal{L} = \mathcal{L}_t + \mathcal{L}_y + \mathcal{L}_r, \quad (5.25) \]

\[ \mathcal{L}_t = \sqrt{\frac{\sum_{i=1}^N \sum_{j=1}^{K_i} (\hat{t}_{j}^i - t_{j}^i)^2}{N \times \sum_{i=1}^N K_i}}, \quad (5.26) \]

\[ \mathcal{L}_y = -\sum_{i=1}^N \sum_{j=1}^{K_i} \sum_{k=1}^{|\mathcal{Y}|} w_k y_j^i \log \hat{y}_j^i, \quad (5.27) \]

\[ \mathcal{L}_r = -\sum_{i=1}^N \sum_{j=1}^{K_i} (w_c r_j^i \log p_j^i + w_r (1 - r_j^i)) \log(1 - p_j^i) 1_{r_j \neq -1}. \quad (5.28) \]

We use weighted cross entropy for type prediction loss $\mathcal{L}_y$. For time prediction, we use the mean squared loss. For the prediction of marker $\mathcal{L}_r$, we use binary cross-entropy. For activities that have a missing marker $r$, we mask the loss via indicator function $1_{r_j \neq -1}$. Similar to the performance prediction function defined in Eq. 5.24, Eq. 5.28 can be replaced by mean squared loss for numerical markers.

\[ ^6 \text{Can be replaced by cross-entropy or regression loss for categorical or numerical } r \]
5.4 Experiments

In the section, we evaluate the proposed model for user activity modeling. First we describe the datasets, then the baseline approaches, and the experimental setups. Later in this section we provide the model evaluation by assessing models’ prediction performances on users’ future activity timings, types, and markers. Ablation study is then performed to determine the importance of our assumptions. 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.

5.4.1 Datasets

We use the following 6 benchmark real-world datasets that contain users’ online activities for the following two popular tasks: (1) student learning and knowledge modeling in online courses, and (2) item recommendation to online users. To ensure that TPPs have a meaningful learning length, users with less than 20 in activities are excluded for this study. An overview of the datasets is given as follows, and some descriptive statistics of these datasets are provided in Table 5.1.

Student Learning Datasets

- **Junyi Academy** \(^7\) 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 math areas labeled by the system are used as tags in this dataset. The outcome of attempting a problem (either correct or incorrect) is used as marker, which is the same for the following two datasets.

- **EdNet** \(^8\) is an AI tutoring platform that assists students in preparing for the TOEIC exam \(^9\). Four flavors of data sets with different levels of details are provided. We specifically use KT4 set, which contains the most information, including student learning interactions with three types of learning materials: questions, lectures, and explanations.

\(^7\)https://pslcdatashop.web.cmu.edu/Project?id=244  
\(^8\)https://github.com/riiid/ednet  
\(^9\)https://www.ets.org/toeic
- **Algebra 2005-2006** is a dataset released for KDD Cup 2010 Educational Data Mining Challenge, which includes students multi-type learning interactions with a computer-aided tutoring system for algebra. The labeled knowledge components are used as tags.

### Recommendation Datasets

- **MovieLens-100k** 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 genres of the movies in this dataset as tags. The ratings of movies, 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** is an e-commerce dataset which contains products in the department of Grocery and Gourmet Food from Amazon. Product categories are used as tags.

- **Yelp** 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.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Activity Types</th>
<th>#. of users</th>
<th>#. of activities</th>
<th>#. of items</th>
<th>#. of tags</th>
<th>Missing Marker</th>
<th>Correct Answer</th>
<th>Avg (std). Inter-arrival Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junyi Problem; Hint</td>
<td>2,063</td>
<td>658K</td>
<td>1,880</td>
<td>8</td>
<td>15.3%</td>
<td>68.8%</td>
<td>21.56 (57.54) mins</td>
<td></td>
</tr>
<tr>
<td>EdNet Problem; Lecture; Explanation</td>
<td>769</td>
<td>282K</td>
<td>12,502</td>
<td>185</td>
<td>71.8%</td>
<td>42.5%</td>
<td>3.54 (29.5) mins</td>
<td></td>
</tr>
<tr>
<td>KDD Problem</td>
<td>574</td>
<td>899K</td>
<td>1,084</td>
<td>112</td>
<td>0%</td>
<td>74.5%</td>
<td>46.17 (85.13) mins</td>
<td></td>
</tr>
<tr>
<td>MovieLens Rating</td>
<td>943</td>
<td>94K</td>
<td>1,682</td>
<td>19</td>
<td>0%</td>
<td>56.0%</td>
<td>5.79 (15.97) days</td>
<td></td>
</tr>
<tr>
<td>Amazon-Grocery Rating</td>
<td>1,342</td>
<td>47K</td>
<td>7,342</td>
<td>126</td>
<td>0%</td>
<td>76.7%</td>
<td>55.40 (29.98) days</td>
<td></td>
</tr>
<tr>
<td>Yelp Rating</td>
<td>1,988</td>
<td>106K</td>
<td>10,230</td>
<td>508</td>
<td>0%</td>
<td>66.0%</td>
<td>22.97 (17.45) days</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.1: descriptive stats of the datasets**

### 5.4.2 Baseline Approaches

In this section, we introduce the baseline approaches. We consider six the state-of-the-art temporal point process models. To better evaluate our proposed model under the
aforementioned settings, we also consider another ten state-of-the-art recent developments of the student modeling and recommendation.

**Temporal Point Process Models.** We consider the following the state-of-the-art temporal point process models as baseline approaches, which will be evaluated on all 6 aforementioned datasets.

- **Poisson Process** [67]: is a classic TPP that models activities by assuming they come in a constant rate.

- **GMHP** [130] is a multivariate Hawkes process model that represents each activity type as a dimension. The dependencies among types are captured by Granger causality graph [29].

- **RMTPP** [32] 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 marker. Following this design, use RMTPP-\(y\) and RMTPP-\(r\) to denote the model that respectively predicts learning material type and grade as markers.

- **ERPP** [129] is a similar approach that models activities and external features via two LSTMs. Similar to RMTPP, we consider two variations when outcome (ERPP - \(r\)) and activity type (ERPP - \(y\)) are used as markers.

- **SAHP** [142] is one of the most recent neural-based Hawkes process model that employs self-attention to capture historical influences of activities on the future ones. This model is 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** [149] 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.

- **DKT** [95] 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** [87] is similar to DKT but uses time differences between exercises as extra features.

- **DKVMN** [141] is based on a Key-Value-Memory-Network that models static knowledge concepts and dynamic students’ knowledge and performances over time.

- **HawkesKT** [117] is a recent first attempt of student knowledge tracing and performance prediction via point processes. Even though Hawkes intensity was adopted, the model is not designed to predict time, nor considers multiple learning material types.

- **MVKM** [145] 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 on the three recommendation datasets.

- **BPRMF** [97] is a matrix-factorization-based method that uses Bayesian Personalized Ranking objective function for the optimization.

- **GRU4Rec** [52] is a session-based sequential recommendation model that utilizes GRU to model the ranking scores.

- **SLRC** [119] is a collaborative filtering model that used Hawkes intensity function to model the temporal aspects of historical activities.

- **TiSASRec** [73] is a sequential model that utilizes self-attention and inter-arrival times between historical activities.
- **KDA** [116] is a Fourier-based temporal method that combined with knowledge graph for item relation modeling.

### 5.4.3 Experimental Setup

For modeling student learning activities, we use 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 [62, 73, 116, 118], 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 [66] in the training process. Each model’s hyper-parameters are tuned separately using the validation set. 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 RMTTPP 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 of the sequence, the best decay rate that leads to the smallest negative likelihood is selected. For THP, the number of attention heads and 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 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, 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 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 $\omega$ 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 $\{1, 2, 4, 8\}$.

5.4.4 Model Evaluation

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

5.4.4.1 Performance Evaluation in Student Learning Datasets

To evaluate activity type and marker predictions, we use 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 5.2. First, we see that compared with all baselines, MoMENt achieves better student grade prediction (i.e. marker prediction) prediction than all baseline approaches, showing the importance of modeling the influences of activity dynamics on marker distribution and vice versa. It can also be observed that recent advances of TPPs such as SAHP, THP and their variants generally outperform other state-of-the-art student models. This suggests that modeling the temporal aspects of student learning via intensity function improves model descriptive power of the data, therefore explain the
### Table 5.2: Performance comparison in Student Learning datasets. Best (bold) and second-best (underlined) are highlighted. Statistically significant differences between MoMENt and the best on confidence levels of 99% and 95% are respectively marked with *** and **.

superiority of these models. In terms of the time prediction, comparing with all TPPs that predicts next activity time, we observe that MoMENt generally achieves lower or comparable time prediction RMSE, even though, in Junyi, the traditional TPP GMHP achieves the smallest RMSE among all models. A possible reason is that in this dataset, students’ learning dynamics are less complex due to fewer number of concepts. Therefore, GMHP’s simple but effective intensity function may capture the data dynamics better, resulting in more accurate time prediction. However, 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. On the other hand, MoMENt is also shown to achieve better type prediction performance, usually by big margins. Combining with the model’s performances in time prediction, it shows that the MoMENt can accurately capture activity arrival patterns, suggesting it can successfully capture activity2activity and marker2activity influences in RAU results in higher model capacity.

To conclude, MoMENt is the first model that can simultaneously predict students’ when to (studying time), how to (learning material type), and how much to (knowledge gain) learn, and consistently outperforms all baseline approaches in terms of all prediction tasks across all datasets, and MoMENt’s superiority suggests the effectiveness and importance of modeling the bi-directional influences between markers and activities.
5.4.4.2 Evaluating MoMENt on Recommendation datasets

Table 5.3 shows the Top-N recommendation performance of all TPPs and recommendation baselines. Two widely used metrics namely hit ratio (HR) and NDCG at 5 and 10 are used as the evaluation metrics for this experiment. As shown in the table, we observe that among all baselines, the state-of-the-art temporal point processes such as SAHP - r and THP - r generally achieve a more competitive performances compared with others, possibly due to an efficient historical activity2activity influence captured by self-attention weights while learning from the intensity function. Recent Top-N recommendation models that employs the intensity functions to capture time dependencies such as KDA and SLRC also achieve promising performances. On the other hand, 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 less promising performances across all recommendation datasets. The combining observation suggests the importance of modeling user temporal dynamics in accurately depicting the dynamics of user behavior. We also see that KDA is shown to be the strongest baseline in Yelp dataset. A possible explanation is that this dataset has a more complex and a larger tag system on all the items (i.e. restaurants) compared with the other two datasets, and KDA’s customized knowledge graph embedding was able to represent items more accurately thus outperforms other baselines.

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 consistent better performance over baseline approaches across all three datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MoMENt</th>
<th>SAHP-r</th>
<th>THP-r</th>
<th>RMTPP-r</th>
<th>ERPP-r</th>
<th>TiSASRec</th>
<th>SLRC</th>
<th>KDA</th>
<th>BPRMF</th>
<th>GRU4Rec</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLens</td>
<td>HR@5</td>
<td>0.428</td>
<td>0.412</td>
<td>0.409</td>
<td>0.387</td>
<td>0.367</td>
<td>0.357</td>
<td>0.387</td>
<td>0.402</td>
<td>0.348</td>
<td>0.345</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.609</td>
<td>0.579</td>
<td>0.570</td>
<td>0.528</td>
<td>0.534</td>
<td>0.529</td>
<td>0.414</td>
<td>0.547</td>
<td>0.484</td>
<td>0.491</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.266</td>
<td>0.256</td>
<td>0.249</td>
<td>0.240</td>
<td>0.239</td>
<td>0.254</td>
<td>0.215</td>
<td>0.254</td>
<td>0.199</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.317</td>
<td>0.310</td>
<td>0.291</td>
<td>0.282</td>
<td>0.279</td>
<td>0.293</td>
<td>0.256</td>
<td>0.308</td>
<td>0.201</td>
<td>0.276</td>
</tr>
<tr>
<td>Amazon-Grocery</td>
<td>HR@5</td>
<td>0.491</td>
<td>0.479</td>
<td>0.481</td>
<td>0.471</td>
<td>0.418</td>
<td>0.386</td>
<td>0.395</td>
<td>0.433</td>
<td>0.410</td>
<td>0.419</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.556</td>
<td>0.538</td>
<td>0.538</td>
<td>0.529</td>
<td>0.549</td>
<td>0.481</td>
<td>0.452</td>
<td>0.533</td>
<td>0.401</td>
<td>0.505</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.389</td>
<td>0.332</td>
<td>0.364</td>
<td>0.354</td>
<td>0.323</td>
<td>0.281</td>
<td>0.252</td>
<td>0.267</td>
<td>0.269</td>
<td>0.280</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.410</td>
<td>0.389</td>
<td>0.401</td>
<td>0.396</td>
<td>0.382</td>
<td>0.321</td>
<td>0.294</td>
<td>0.398</td>
<td>0.333</td>
<td>0.316</td>
</tr>
<tr>
<td>Yelp</td>
<td>HR@5</td>
<td>0.463</td>
<td>0.401</td>
<td>0.392</td>
<td>0.408</td>
<td>0.413</td>
<td>0.284</td>
<td>0.402</td>
<td>0.422</td>
<td>0.380</td>
<td>0.280</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.626</td>
<td>0.567</td>
<td>0.554</td>
<td>0.549</td>
<td>0.537</td>
<td>0.535</td>
<td>0.609</td>
<td>0.617</td>
<td>0.598</td>
<td>0.460</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.299</td>
<td>0.267</td>
<td>0.250</td>
<td>0.267</td>
<td>0.259</td>
<td>0.278</td>
<td>0.264</td>
<td>0.303</td>
<td>0.258</td>
<td>0.175</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.393</td>
<td>0.345</td>
<td>0.317</td>
<td>0.324</td>
<td>0.348</td>
<td>0.375</td>
<td>0.378</td>
<td>0.381</td>
<td>0.377</td>
<td>0.233</td>
</tr>
</tbody>
</table>

Table 5.3: Performance comparison in recommendation datasets. Best (bold) and second-best (underlined) are highlighted.
5.4.5 Ablation Study

In this ablation study, we assess the importance of activity2marker influence since it has been largely neglected by the literature. We specifically consider MoMENt-t, MoMENt-y and MoMENt-ty 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 Knowledge Tracing domain and all datasets in the recommendation domain only have one type of activities, only the results of MoMENt-t are meaningful and presented.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MoMENt</th>
<th>MoMENt-t</th>
<th>MoMENt-y</th>
<th>MoMENt-ty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Junyi</td>
<td>marker AUC</td>
<td>0.837</td>
<td>0.826</td>
<td>0.831</td>
<td>0.816</td>
</tr>
<tr>
<td></td>
<td>marker ACC</td>
<td>0.855</td>
<td>0.835</td>
<td>0.840</td>
<td>0.798</td>
</tr>
<tr>
<td></td>
<td>type AUC</td>
<td>0.631</td>
<td>0.524</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>type ACC</td>
<td>0.935</td>
<td>0.883</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>time RMSE</td>
<td>9.023</td>
<td>-</td>
<td>9.904</td>
<td>-</td>
</tr>
<tr>
<td>EdNet</td>
<td>marker AUC</td>
<td>0.898</td>
<td>0.873</td>
<td>0.861</td>
<td>0.855</td>
</tr>
<tr>
<td></td>
<td>marker ACC</td>
<td>0.909</td>
<td>0.875</td>
<td>0.863</td>
<td>0.856</td>
</tr>
<tr>
<td></td>
<td>type AUC</td>
<td>0.704</td>
<td>0.677</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>type ACC</td>
<td>0.966</td>
<td>0.959</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>time RMSE</td>
<td>5.890</td>
<td>-</td>
<td>8.226</td>
<td>-</td>
</tr>
<tr>
<td>Algebra</td>
<td>marker AUC</td>
<td>0.863</td>
<td>0.849</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>marker ACC</td>
<td>0.887</td>
<td>0.852</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>time RMSE</td>
<td>9.111</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.4: Ablation study in Knowledge Tracing datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Metric</th>
<th>MoMENt</th>
<th>MoMENt-t</th>
</tr>
</thead>
<tbody>
<tr>
<td>MovieLense</td>
<td>HR@5</td>
<td>0.428</td>
<td>0.398</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.609</td>
<td>0.577</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.266</td>
<td>0.236</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.317</td>
<td>0.272</td>
</tr>
<tr>
<td>Amazon-Grocery</td>
<td>HR@5</td>
<td>0.491</td>
<td>0.467</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.556</td>
<td>0.530</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.389</td>
<td>0.375</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.410</td>
<td>0.383</td>
</tr>
<tr>
<td>Yelp</td>
<td>HR@5</td>
<td>0.463</td>
<td>0.403</td>
</tr>
<tr>
<td></td>
<td>HR@10</td>
<td>0.626</td>
<td>0.607</td>
</tr>
<tr>
<td></td>
<td>NDCG@5</td>
<td>0.299</td>
<td>0.258</td>
</tr>
<tr>
<td></td>
<td>NDCG@10</td>
<td>0.393</td>
<td>0.302</td>
</tr>
</tbody>
</table>

Table 5.5: Ablation study in recommendation datasets.

The results of student learning modeling and Top-N recommendation are respectively shown in Table 5.4 and Table 5.5. In the student learning datasets where the goal was to
predict student future learning time, type and outcome (i.e. marker), the result in Table 5.4 shows that MoMENt outperforms variants across all datasets in all predictions. Comparing with the variants, we see that variant MoMENt-ty achieves the lowest performance of marker prediction. This demonstrates that simultaneously modeling time, type and marker are essential to capture the complex dynamics of student learning in terms of time, type, and learning outcome predictions. We also notice that in EdNet, masking type (MoMENt-y) results in a less competitive marker prediction performance than masking time (i.e. MoMENt-t), whereas in Junyi, this observation is reversed. This is potentially related to the more types of learning materials that are provided in EdNet, which lead to more complex knowledge dynamics in students, e.g., when students switch from one learning material type to another.

In the task of Top-N recommendation where models generate N top items recommendations based on the rank of item likelihood of having positive (i.e. a positive rating), we observe that the proposed MoMENt outperforms the variant MoMENt-t usually by large margins. This highlights again the significance of activity time in representing user activity dynamics and marker distributions (positive or negative ratings). For example, a user’s future interactions with items are not only associated with the items they interacted with from the past, 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 Ch. 5.4.4.

5.4.6 Interpretation and Analysis

In this section we present meaningful interpretations MoMENt can provide, where two case studies from the two application domains are respectively given. Within each case study, the analysis is divided into two part. We first demonstrate MoMENt’s ability in capturing a fine-grained interpretation of user-system relation over time. Furthermore, an analysis shows the temporal patterns of user interactions is provided, where we analyze how users’ historical behavior can have an influence on users’ future activities.
5.4.6.1 Understand Student Learning Dynamics in Online Courses

Modeling students’ learning activities while quantifying student knowledge is an essential task in the education domain. 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. The aim of this section is to demonstrate how to use MoMENt’s interpretation to fill this gap. Consider an online course, suppose there are $C$ knowledge concepts covered by the course learning materials. Then, course concepts and student knowledge at step $j$ can be represented by the static key matrix $M^k$ and dynamic value matrix $M^v_j$.

Knowing students’ knowledge of course concepts can be beneficial for the improvement of teaching or tutoring quality, where students’ strengths and weakness can be better assessed by the tutoring systems. In order to obtain students’ knowledge, we pass the learned value matrix $M^v_j(z)$, namely the $z$-th row of the value matrix $M^v$ at step $j$ through Eq. 5.22-5.24 using the learned parameters after training. The resulting value can be interpreted as 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 student’s changing knowledge over time. A sample student from Junyi Academy and their evolving knowledge states of 3 latent concepts over 20 practicing steps are given in Fig. 5.4.

![Figure 5.4](image)

Figure 5.4: Changing knowledge states of a sample student from Junyi dataset over 20 steps of practice (x-axis) w.r.t concepts $C_1$, $C_2$, and $C_3$ (y-axis). The tags of the problems over 20 steps are given below the parentheses (e.g. multiplication problems for the last 2 steps). The markers $r$ are given in the top row, with red, green, and yellow respectively representing wrong answers, correct answers, and hint-checking. The shades of cells in the three bottom rows represent the degrees of student’s understanding, ranging from yellow (low) to purple (high).

As it is shown in the figure, we see that the student practiced some mean and median
problems and got the wrong answers over these problems (step 1-5 marked in red). During this practice, we noticed that student’s understanding of concepts 2 and 3 started 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 (step 6-9 in purple), 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 to gain a better understanding of these concepts, to later obtain correct answers (step 10-13 in pink) and have a much faster knowledge gain. Later, when the student moves on to another type of problems (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. It may also explain that the student 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 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.

Furthermore, 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 dependencies within the trajectory of student learning, or in other words the influences of student historical learning on their future learning activities. 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 [135] and the modeling of knowledge forgetting [109]. To examine student learning dynamics in continuous time, we present student’s learning intensities as a function of time while they are interacting with different learning materials (i.e. answering problems and checking hints). This is presented in the upper plot in Fig. 5.5. Furthermore, to understand the influence of historical learning activities to the future (e.g. checking a hint at step 6 and it’s association to getting a correct answer in later steps), we compute the influence of each historical $j$-th activity to the later $i$-th activity where $i > j$, represented by cell at $j$-th column $i$-th row. More specifically, such influence is computed via the Hawkes...
Figure 5.5: Temporal patterns of student learning represented by learning intensity (upper) and influence of historical learning on the future (lower). Upper: intensities of student learning dynamics ($y$-axis) as a continuous function of time $t$ ($x$-axis), and the colored-shaded areas have the same meaning as the color-coded markers. Lower: influences of historical activities on the future ones for the same sample student during practice. Cell in $j$-th row $i$-th column represents the influence of activity at step $j$’s influence 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, green, and yellow respectively represent wrong answer, correct answer and hint-checking.

component in Eq. 5.14, where past $j$-th activity’s influence to the future $i$-th activities can be interpreted as the expected intensity at $i$-th activity that were influenced by the $j$-th activity [49]. This is presented as in the lower plot in Fig. 5.5.

As it is shown in Fig. 5.7, student’s first a few steps of practice have a high and lasting effect to the later activities (darker cells under the diagonal in steps 2-9 in the green box). A possible explanation is that getting wrong answers in a row (step 1-5) made student to realize that they lack some knowledge in answering these problems. Therefore the student started to check the hints (step 6-9) intensively (steep blue line in the first red area in the upper plot) trying to understand why the previous a few answers are wrong. And as the student practicing more, the influences from the past wrong answers on the later activities started to decay, possibly because these previous problems started to have less and less association with student’s later practice (fading shade on the column-wise). Another interesting pattern can
be found in student’s later steps of learning, where student’s learning activities are shown to have a strong association with their most recent prior activity (e.g. step 10-13 in the blue box). During these steps, we see that the student got one correct answer after another with a steady pace (steady red line and spaced out red dots in green area in the upper plot). One explanation is that answering a question correct has small influence on student’s next a few activities, probably because the student does not have the urge to correct themselves by practicing more or checking hints. Rather, the student is shown to try the next ones using similar pace and got a good understanding of the problems (suggested by fast increasing knowledge in concept 2), then switched to a different type of problems after step 13.

5.4.6.2 Understanding User Movie-watching Dynamics

In the application of recommendation, we also start our analysis of user-system relationship captured by MoMENt, or in other words, user’s evolving relationship to latent components of items under this concept.

![Figure 5.6:](image)

Figure 5.6: Upper: evolving user’s interest w.r.t 3 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 of movies where a white cell in a row represents the movie is of the corresponding row of genre while black represents the opposite. Top 10 genres were shown in this plot.

In this section, we consider one sample user from the MovieLense dataset for visualization and analysis, as shown in Fig. 5.6. Similarly, we use the z-th row of the value matrix $M^v$ at step $j$ and the learned parameters after training to obtain the evolving relation of
user to to each component $C_z$, which under this setting can be interpreted as user’s interests of different movie components. Ground truth of markers are again provided in the top row of this plot, with red and green respectively representing negative and positive ratings. The genres of the movies are represented as a heat-map with a binary color scheme (lower plot in Fig. 5.6), where a bright cell at $j$-th column and $g$-th row represents that the movie at $j$-th step is a movie of genre in $g$-th row, and a dark cell represents that it is not. For example, in the given example, the last movie is of the genres of fantasy and Sci-Fi. As shown in Fig. 5.6, the user tried a few Crime movies (step 24-37) between fantasy movies. While the user was watching these movies, their interests of component 3 decreased and once the user switched back to the fantasy movies it keeps being low. This suggests that concept 3 is likely to be mostly associated with some common elements in crime movies such as heist or robbery, and when the user was exposed to such elements repetitively, their general interest of component 3 decreased. This observation matches to the ground truth where some negative ratings were given to Crime movies during these steps. On the other hand, components 1 and 2 may be more related with elements in Fantasy movies (e.g. magic and fairy tale) which matches their fast increase while the user was watching this genre and giving positive ratings for these movies.

To analyze the dynamics of user-system interactions, similar to the analysis in the previous domain, 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 Fig. 5.7. In the same manner, markers i.e. positive and negative ratings are respectively represented using red and green. As shown in the figure, examples of two different influence patterns can be observed and presented in green and blue boxes. For example, the first several fantasy movies the user watched are shown to have lasting and strong influence on user’s future choices of movies to watch (green box). e.g. watching Harry potter 1 may consequently motivate the user to watch the next a few movies in the series to complete the story. During this time, we also see a growing trend of intensity (red line in upper plot) when the watching experience was good (green area). One explanation is that the user enjoyed some good movies from the past therefore is willing to try more within a shorter period of time (i.e. bursts of activity clusters). More interestingly, this phenomenon also is in line with the study that reveals the association between binge-watching-like behavior and the dopamine released by human brains [42], which can possibly explain the high fast-growing
intensity and interests (as shown in Fig. 5.6) during these bursts of activities. Another interesting pattern is shown in the blue box where activities are mostly influenced by the most recent ones from the past, suggested by the darker area only close to the diagonal. While the user was watching a movie during this time, we see that the movie user watched right before usually has a much stronger effects than the ones user watched further back in time. Combining with the observation from the upper plot where the intensity of watching decreased and mostly negative ratings were given during this time, a possible explanation is that after watching a crime movie that the user did not enjoy, the user is less motivated to watch more. In other words, these movies have less influence on user’s future choices of what movies to watch.

In summary, this section shows MoMENt’s effectiveness in providing meaningful explanation of when and how user interact and the associations between their interaction dynamics, continuous time intensity, and their relation to the system.

![Figure 5.7: Dynamics of user moving-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 j-th row i-th column represents the influence of 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.](image-url)
5.5 Conclusion

To answer RQ3, in this work, we proposed a novel marked point process model called MoMENt, that captures the fine-grained and interpretable bi-directional influences between activity dynamics and activity markers that have been overlooked by the literature. 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 outcome dynamics. We applied MoMENt to the problems of student activity modeling in online education systems, as well as Top-N recommendations. We demonstrated its effectiveness in predicting activity time, type and marker, by comparing it with various state-of-the-art TPPs and domain models via extensive experiments across six real-world datasets. In our ablation study, we showed the importance of study time vs. type modeling in predicting student performance in different systems and datasets. Our case study reveals interpretable representations of user-system relation over time, and provides meaningful insights on how users future interactions are influenced by their past experience.
CHAPTER 6
Modeling Granger Causality in Multivariate Temporal Point Processes

6.1 Introduction

Real-world data usually contains massive amount of complex and interdependent events in real time that reflect user preference and motivations. Examples include students practicing different types of problems in online education systems to learn and hone their knowledge, or user watching various categories of TV programs to relax or get access to new information. These multi-type interactions can be viewed as multivariate point processes, which are a special type of stochastic process where each event (i.e. a user activity) is characterized by its type (i.e. dimension) and its timestamp.

Modeling and interpreting the dynamics of user interactions is an important but challenging task. In many real-world applications, it is not only important to know when a user will interact but also why they are initiating this type of interaction. To answer these questions, in this work, we aim to uncover causal relation among event types from multi-type user interaction data in continuous time. Particularly, in this work, we consider Granger causality [46] that is defined according to predictability. Despite the importance of the problem, the literature on modeling Granger causality in point processes is still extremely limited, owing in part to the inherent difficulty of uncoupling causality between dimensions while the modeling of event sequences takes place in continuous time. In the few previous works, the emphasis was mostly on the modeling of multivariate Hawkes processes, one of the most common families of point processes, employing linear kernel functions [16, 37, 130]. This line of methods, despite their great interpretations in explaining data dynamics, are largely restricted by their assumption that data is self-exciting and the intensity of activities are linearly additive, i.e. the past activities are linearly and additively triggering the arrivals of future activities. As a result, other types of dynamics such as self-inhibiting (i.e. new arrivals are inhibited or regularized by past ones) or more complicated data dynamics such as those are nonlinearly additive can not be properly handled.
Recently, a new family of point processes, known as neural point processes, has been developed to leverage recurrent neural networks (RNNs) to represent the underlying dynamic of the data more flexibly and accurately [32, 82, 127, 149]. These models, however, by the nature of RNNs are hard to interpret [131]. Particularly, the interpretation of why the activities are arriving in such manner and how dimensions of activities can influence each other is lost in the modeling. More importantly, the challenge of disentangling the historical influences of all dimensions has not been heavily addressed. Current neural point process models usually use one RNN to represent the entire history of the processes regardless of the dimension, which consequently compromises their interpretation even further.

As another limitation, even though the definition of Granger causality in point processes was given based on the traditional intensity function, it has not been properly defined under the general framework of neural point processes. The same framework has been adopted to neural point process models directly from the traditional point process models. As a result, for the task of identifying Granger causality among dimensions, the formulation of intensity function even in neural point process models is largely constraint by its linearity and rigid format retained from traditional point process models. For example, Zhang et al. recently proposed the first Granger causality neural point process model CAUSE [143]. However only one GRU is used to model the history of all dimensions, making it harder to differentiate the histories from different dimensions. Furthermore, similar to traditional Hawkes processes, an intensity function modeled by linearly additive base functions has been used, to fit into the definition of Granger causality given in traditional models [37, 130]. The question of how to model Granger causality in neural multivariate point processes using a more flexible and general framework is not studied.

To address the aforementioned limitations, we propose REACT (REcurrent GrAnger Causality in poinT process), a novel framework that captures Granger causality in multivariate point processes modeled via recurrent neural network. More specifically, we represent each dimension using one RNN while maintaining their dependencies, enabling us to decouple Granger causality from one dimension to another. Furthermore, we discover Granger causality among dimensions considering a general framework using RNN that is not limited to intensity function formulations, allowing for a more flexible representation in neural point processes without sacrificing the interpretation.
6.2 Background

In this section, we provide some basic concepts from measure and probability theory as well as stochastic filed to help build the basis of the proposed method.

6.2.1 Prior on Measure and Probability Theory

The following notions and concepts are primarily obtained and summarized from [35] \(^{14}\). In a probability space \((\Omega, \Sigma, P)\), the set \(\Omega\) is the set of all outcomes of “probability experiment”, and it is just a set of elements of \(w\) and it is called the sample space.

A \(\sigma\)-algebra generated by \(\Omega\), denoted as \(\Sigma\), is a collection of possible events from the experiment at hand. In other words, a \(\sigma\)-algebra is a mathematical model of a state of partial knowledge about the outcomes. Informally, if \(\Sigma\) is a \(\sigma\)-algebra and \(A \in \Omega\), we say that \(A \in \Sigma\) if we know whether \(w \in A\) or not.

\((\Omega, \Sigma)\) is a measurable space if \(\Omega\) is a set, and \(\Sigma\) is a non-empty \(\sigma\)-algebra of subsets of \(\Omega\). A measure defined on \((\Omega, \Sigma)\) is a non-negative function \(\mu\) defined on \(\Sigma\) iff:

1. \(0 \leq \mu(A) \leq \infty\) for all \(A \in \Sigma\)
2. \(\mu(\emptyset) = 0\)
3. (\(\sigma\)-additive) \(\mu(\bigcup A_i) = \sum_i \mu(A_i)\)

Let \((\Omega, \Sigma, P)\) be a probability space, a stochastic process is a collection of random variables \(\{X_i\}_{i \in \mathcal{I}}\) for a index set \(\mathcal{I}\) (usually real line \(\mathbb{R}_+\)). If \((\Sigma_i)\) is a family of \(\sigma\)-algebra on \(\Omega\) such that \(\Sigma_i \subset \Sigma_{i+1} \subset \Sigma\) for all \(i\), then \(\mathcal{H} = (\Sigma)_i\) is called a filtration of \((\Omega, \Sigma, P)\), or write as \(X\) is \(\mathcal{H}\)-measurable. Then, \(\mathcal{H}_i = \sigma(\{X_j | j \in \mathcal{I}, j \leq i\})\) is called a natural filtration, which is the simplest filtration associated with the stochastic process \(\{X_i\}\) with the records of the past of the process. For this reason, the natural filtration \(\mathcal{H}_i\) is often viewed as the “history” of the process. In point processes (more in Ch. 6.2.2), natural filtration is usually referred to as history and denoted as \(\mathcal{H}_t\) to represent the history of a process up to time \(t\).

\(^{14}\)Notations are sometimes altered for the consistency of the definitions in both measure theory and point processes.
6.2.2 Elementary Concepts of Multivariate Point Process

A univariate point process \[ X(t) : t \geq 0 \] such that \( X(0) = 0 \) that is almost surely finite. It is usually defined by its conditional intensity function \( \lambda^*(t|\mathcal{H}_t) \) or for simplicity more often denoted as \( \lambda^*(t) \).

A \( D \)-dimensional point process is a point process that consists of both time \( t \) and dimension \( y \), i.e. \( X = \{(t_1, y_1), \ldots, (t_K, y_K)\} \), where \( y_j \in [1, \ldots, D] \). An equivalent view of a \( D \)-dimensional point process is \( D \) coupling point process \( X = \{X_1, \ldots, X_D\} \), where each subprocess \( X_d, d \in [1, \ldots, D] \) (usually called the \( d \)-th dimensional process) is a univariate point process uniquely characterized by time, which can be then defined by its own conditional intensity function, denoted as \( \lambda^*_D(t) \).

One important property of point process is that it is a submartingale thus can be decomposed into a zero-mean martingale and a compensator (later in this section).

Since this decomposition serves the building block of the definition Granger Causality in point processes, we briefly introduce martingales, submartingales, and compensators first in this section.

**Definition 6.2.1.** Let \((\Omega, \mathcal{H}, P)\) be a probability space and let \( \mathcal{H} \) be a filtration of the measurable space \((\Omega, \Sigma)\). Then the process \( X_n \) is a martingale wrt filtration \( \mathcal{H} \) if it satisfies the following properties:

1. \( X_n \) is \( \mathcal{H} \)-measurable
2. \( E(|X_n|) < \infty \)
3. \( E(X_n|\mathcal{H}_m) = X_m \) almost surely for all \( m \leq n \). If replacing the third condition by the inequality \( E(X_n|\mathcal{H}_m) \geq X_m \) almost surely for all \( m \leq n \), then the process is a submartingale [35].

Recall a point process can be defined by its conditional intensity function \( \lambda^*(t) \), then

**Definition 6.2.2.** A compensator of a point process is defined as the integral of its conditional intensity function as follows:

\[
\Lambda(t) = \int_i^t \lambda^*(s)ds, \tag{6.1}
\]

This definition can be intuitively interpreted as the conditional mean of \( X(t) \) given its
history (i.e. natural filtration \( \mathcal{H}_t \)) [7].

Now, Let \( X(t) \) be a stationary \( D \)-dimensional point process, the following gives a proposition of an important property of point process:

**Proposition 6.2.1.** \( X(t) \) is submartingale.

The proof of this proposition can be found in [7] which verifies the 3 properties outlined in the definition of martingales.

Then, ensured by the Doob-Meyer decomposition theorem [114], the following proposition can be obtained:

**Proposition 6.2.2.** A submartingale \( X(t) \) can be decomposed into a zero-mean martingale \( M(t) \) and a unique \( \mathcal{H}_t \)-predictable increasing process \( \Lambda(t) \), as follows:

\[
X(t) = M(t) + \Lambda(t), \quad \text{s.t. } E(M(t)) = 0. \tag{6.2}
\]

### 6.3 Proposed Model

In this section, we will introduce our proposed model REACT (REcurrent GrAnger Causality in pointT process) that not only flexibly characterizes complex dynamics of event sequences (Ch. 6.3.1) but also automatically captures Nonlinear Granger Causality among different types of events (Ch. 6.3.2).

#### 6.3.1 Modeling Multivariate Point Processes via RNNs

Suppose we are given a collection of \( N \) point processes \( \mathcal{X} = \{X^1, \ldots, X^N\} \), where each \( X^i = \{(t^i_j, y^i_j)\}_{j \in [1, K^i]} \) is a multivariate point process of length of \( K^i \), with \( t^i_j \in \mathbb{R}_+ \) and \( y^i_j \) respectively representing the timestamp and and the dimension (i.e. type) of the \( j \)-th event.

As introduced previously, for a point process, the distributions of inter-arrival time within the process can be characterized by its conditional intensity function \( \lambda^i(t|\mathcal{H}_t) \). On the other hand, by definition, for multivariate point processes, e.g. multivariate Hawkes processes, the intensity function of dimension \( d \), \( \lambda^i_d(t) \) can be defined as a kernel function \( \phi(t) \), which traditionally has been modeled by exponential function [146] or Gaussian function [130]. To relax such rigid parameterization, neural point process models instead model...
conditional intensity functions via RNNs analogically. Despite its flexibility in modeling complex dynamics of activities, using a RNN to model multivariate point processes makes it difficult disentangling the historical influence of one dimension on another, which is not sufficient for Granger causality modeling in multivariate point processes.

Motivated by this limitation, we propose to model each \(d\)-th dimensional point process in \(\mathcal{X}\) using one RNN for all \(d \in D\). A challenge remaining to allow relation among each pair of dimensions to be captured while each dimension is modeled by a separate RNN.

To deal with this challenge, inspired by the formulation of multivariate Hawkes process, we propose to represent the input of point processes as follows: for each multivariate point process \(X\), based on the definition of multivariate point process stated in Ch. 6.2.2, it can be further decomposed into \(D\)-coupling uni-variate point processes \(X = \{X_1, ..., X_D\}\). Each \(d\)-th process then can be represented by the timestamps of all its events belong to dimension \(d\), or in other words, \(X_d = \{t_{d,1}, ..., t_{d,K_d}\}\) represents the collection of activities that are of type \(d\), where \(K_d\) is the total number of activities in dimension \(d\). This decomposition allows us to represent the dynamic of the multivariate point process as \(D\) sequences of inter-arrival times between current time and the history of each sub uni-variate point process. More formally, the following is used as input:

\[
\forall j \in [1, K], \ x_j = (t_j - t_{d,j} | t_{d,\tau} \leq t_{d,j} < t_j, \ \forall d \in [1, ..., D], \ \forall t_{d,\tau} \in \mathcal{H}_{d,j}) \in \mathbb{R}^D \quad (6.3)
\]

Intuitively, the above format represents the collection of the time differences between the current time \(t_j\) and the time of the most recent activity \(t_{d,j}\) from each dimension \(d\). Then the final input is of the format:

\[
x = (x_j^i)_{i,j} \in \mathbb{R}^{N \times K \times D}, \quad (6.4)
\]

with \(N, K, D\) respectively representing the size, length, and the number of dimensions of \(\mathcal{X}\).

Note that this input structure has an intuitive meaning that is similar to the traditional \(D\)-dimensional Hawkes process, whose intensity function of dimension \(d'\) is a function of
inter-arrival time \( t - t_{d,j} \), usually defined as follows:

\[
\forall d' \in [1, ..., D], \quad \lambda^*_d(t) = \mu_i + \sum_{d=1}^{D} \sum_{t_{d,j} < t} \phi_{d,d'}(t - t_{d,j}).
\]  \hfill (6.5)

We see that similar to this formulation, the inter-arrival times between current time and each previous activity time from each dimension is preserved in the input, thus allows us to model each dimension using one RNN without losing the dimension dependency.

To this end, considering LSTM’s effectiveness in modeling complex dependencies, we choose to use an LSTM to model dimension \( d \) of the multivariate point process given \( j \)-th step input \( x_j \) as follows:

\[
i_{d,j} = \sigma(W_{di}x_j + U_{di}h_{d,f-1}),
\]

\[
f_{d,j} = \sigma(W_{df}x_j + U_{df}h_{d,f-1}),
\]

\[
c_{d,j} = f_{d,j} \odot c_{d,j-1} + i_{d,j} \odot \tanh(W_{dc}x_j + U_{dc}h_{d,j-1}),
\]

\[
o_{d,j} = \sigma(W_{do}x_j + U_{do}h_{d,j-1}),
\]

\[
h_{d,j} = o_{d,j} \odot \tanh(c_{d,j}),
\]

(6.6) \hfill (6.7) \hfill (6.8) \hfill (6.9) \hfill (6.10)

where \( i_{d,j}, f_{d,j}, o_{d,j} \) are respectively input, forget and output gates of this \( d \)-th dimensional LSTM at step \( j \) that control how the cell \( c_{d,j} \) is updated and transferred to hidden state \( h_{d,j} \) from step \( j - 1 \) recursively. Since the inter-arrival times from all dimension are encoded in input \( x_j \), the embedding \( h_{d,j} \) represents the memory of the influence from the timings from all previous activities from all dimensions. Since \( h_{d,j} \) contains information of the history \( \mathcal{H} \), naturally, the intensity function of dimension \( d \) can be modeled as follows:

\[
\lambda_d(t|\mathcal{H}_x) = \text{relu}(b_d + \sum_{d'} \alpha_{dd'}(h_{d',j}) \exp \{- \beta_{dd'}(h_{d',j})x_j\}),
\]

(6.11)

\[
\beta_{dd'}(h_{d',j}) = (W_{\beta_{dd'}}h_{d',j})_+,
\]

(6.12)

\[
\alpha_{dd'}(h_{d',j}) = \tanh(W_{\alpha_{dd'}}h_{d',j}).
\]

(6.13)

For the time prediction, we use the following equations:
\[ f^*(t) = \lambda_d^*(t) \exp \left( - \int_{x_j}^t \lambda^*(\tau) d\tau \right), \quad (6.14) \]

\[ g_{j+1} = \tanh(w_g^T \log(f^*(t_j)) + b_g), \quad (6.15) \]

\[ \dot{x}_{j+1} = \text{relu}(w_t^T g_{j+1} + b_t). \quad (6.16) \]

### 6.3.2 Granger Causality Selection in Multivariate Point Process

In Ch. 6.3.1, we proposed a neural multivariate point process framework to flexibly represent the dynamics from all dimensions of the point processes. The goal of this section is to identify, for each \( d \in \mathcal{D} \), a subset \( \mathcal{D}' \subset \mathcal{D} \) such that each \( d' \in \mathcal{D}' \) Granger causes \( d \). Before embarking on the modeling of Granger causality in neural multivariate point process model, it is necessary to go over the definition of Granger (Non-) Causality in a general stochastic process, which is given as follows [44]:

**Definition 6.3.1.** Stochastic process \( X_{d'} \) does not Granger cause process \( X_d \) iff

\[ E(X_d(t + \Delta t)|\mathcal{H}_t) = E(X_d(t + \Delta t)|\mathcal{H}_{t,-d'}), \quad \forall t, \forall \Delta t \geq 0 \quad (6.17) \]

Intuitively, the condition on the right hand side of iff states that the mean intensity of \( X_d \) remains the same with (i.e. \( \mathcal{H}_t \)) or without the history of dimension \( d' \) (i.e. \( \mathcal{H}_{t,-d'} \)). This definition naturally leads to the idea that \( X_{d'} \) from all its history does not bring any information about \( X_d \) all the time (i.e. \( E(X_d(t + \Delta t)|\mathcal{H}_t) = E(X_d(t + \Delta t)|\mathcal{H}_{t,-d'}) \)). In the context of multivariate point processes, the above definition is equivalent to saying that if the history of dimension \( d' \), which is uniquely characterized by its intensity \( \lambda_{d'}(t) \), is locally independent from dimension \( d \), then dimension \( d' \) does not Granger causes dimension \( d \), otherwise it does.

Revisiting the neural multivariate point process model proposed in the previous section (Ch. 6.3.1), the set of input weights

\[ W_d = (W_{d_{ii}}, W_{d_{if}}, W_{d_{ic}}, W_{d_{io}})^\top \in \mathbb{R}^{D \times 4H} \]

controls how the inter-arrival times \( x_j \) influence the current state of dimension \( d \) by controlling input, forget, output gates and cell states. Now, based Definition 6.3.1, to capture
Granger non-causality of dimension $d'$ on $d$, a natural idea is to rule out the influence of history of $d'$ on $d$ by letting the $d'$-th row of the weight matrix $W_d$ to be all zero. To do so, particularly, we apply the $\mathcal{L}_{2,1}$ norm on matrix $W_d$ for all $d$, namely reinforcing rows of matrix $W_d$ to be all zeros. Intuitively, the inter-arrival times between current activity and the most recent one from dimension $d'$ does not influence the hidden state and thus the intensity in dimension $d$, thus does not Granger cause dimension $d$.

This allows us to further extend Definition 6.3.1 based on the proposed framework in Sec 6.3.1 as follows:

**Definition 6.3.2.** Considering $D$-dimensional point process $X = \{X_1, ..., X_D\}$, $d'$-th dimensional point process $X_{d'}$ does not Granger cause $d$-th dimensional point process $X_d$ iff the conditional intensity function $\lambda_d$ is $\mathcal{H}_{t,-d'}$-measurable for all $t$, or in other words, it is not influenced by $x_{d',j} \in x_j$ (Eq. 6.3) for all $j > 0$ or equivalently, with $W_d(d') = 0$ (Eq. 6.18).

More formally, we provide the following sketch of proof to show the above structure is consistent with the definition of Granger causality in general stochastic processes (Definition 6.3.1), and hence consistent with multivariate point processes:

**Proof.** We know from Proposition 6.2.1 that a multivariate point process $X$ is submartingale, so by Proposition 6.2.2, $X_d$ can be decomposed as $X_d = M_d + \Lambda_d$ wrt filtration $x_j$ for all $j$, with $M_d$ being zero-mean. Then according to Definition 6.3.1, if $X_{d'}$ does not Granger cause process $X_d$, then the above decomposition of $X_d(t)$ wrt $\mathcal{H}$ and $\mathcal{H}_{t,-d'}$ are identical. In other words, given a multivariate point process $X$ with history, $X_{d'}$ does not Granger Cause $X_d$ if the history of dimension $d'$, which is characterized by its intensity function $\lambda^*(t)_{d'}$ does not bring any change to the compensator or equivalently the intensity of dimension $d$ $\lambda^*(t)_d$, ensured by the transition equation in definition 6.2.2. □

To this end, the loss of REACT for modeling $X = \{X^1, ..., X^N\}$ can be defined as follows, that is consists of a Mean Squared Errors (MSE) loss for time prediction and the
loss for regularizing Granger causality:

\[ L = L_{\text{time}} + L_{\text{GC}} \]

\[ = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{D \times K} \sum_{j=1}^{K} \sum_{d=1}^{D} (x_{jd}^i - \hat{x}_{jd}^i)^2 + \rho \sum_{d=1}^{D} L_{2,1}(W_d) \]

(6.18)

(6.19)

6.4 Experiments

6.4.1 Datasets

In this work, we consider both real-world datasets and simulated datasets.

6.4.1.1 Simulated Datasets

To check models’ performances in modeling different data dynamic, we particularly consider two common real-world scenarios namely self-exciting and self-inhibiting, generated as follows:

Self-exciting. First we generated self-exciting processes [49] defined by the following intensity function: \( \lambda_i(t) = \mu_i + \sum_{t_j < t} \alpha_{ij} \beta_{ij} \exp(-\beta_{ij}(t - t_j)) \) More specifically, we generated 500 sequences with \( D = 10 \) namely 10 dimensions, with base rate generated from Uniform \((0, 0.05)\), decay generated from Exp \((0.05)\), self-exciting coefficient matrix is generated from Uniform \((0, 1)\) on the diagonal to simulated within-dimension self-excitement and 4 random off-diagonal cells to simulate between-dimension self-excitation.

Self-inhibiting. We also consider self-inhibiting processes [57], with the intensity defined as follows: \( \lambda_i(t) = \exp(\mu_i t - \sum_{t_j < t} \alpha_{ij}) \). We sampled 500 sequences with \( D = 10 \) namely 10 dimensions. In inhibiting matrix \([\alpha]_{ij}\), we first sample diagonal from Uniform \((0, 0.01)\) to simulate within-dimension self-inhibition and then randomly generated 5 off-diagonal cells to simulate between dimension dependencies.

6.4.1.2 Real-world Datasets

We also consider the following two real-world datasets:

IPTV [79] is a benchmark dataset for the literature of causality TPPs. This dataset contains 1473 sequences of user watching behavior of IPTV, which is an English TV channel in
Shanghai China. There are 16 categories of TV in this data such as sports and news, and the watching records of each category is modeled as a dimension, which leads to 16-dimensional TPP. 

Junyi is collected from a Chinese e-learning website Junyi Academy, which contains the trace data of students’ problem-practicing activities from a math module. 8 domains of problems (e.g. algebra) are provided in Junyi and each of them is modeled as a dimension which leads to a 8-dimensional TPP.

6.4.2 Baseline Approaches

- MLE-SGLP [130] is a classic Hawkes process model that uses Gaussian kernel functions and Group Lasso to represent Granger Causality among activity types.

- RPPN [128] is a recurrent point process model that consider auxiliary time series features which are modeled by two independent LSTMs. Self-attention was added to provide quasi-causality interpretation among types of activities.

- CAUSE [143] proposed to use Attribution Methods to capture causality among dimensions in Hawkes modeling. Causality is captured by regularizing baseline function to be close to zero

- THP [16] is the most recent Hawkes process models that capture granger causality that measures causal relations among types considering the scenario where sequences are related when signals can propagate among them.

6.4.3 Experiment Setup

Data splitting In each dataset, 80% of sequences and their activities are selected for training, 10% are randomly selected for validation, and the rest 10% are used for testing. For the training, we use Adam [66] optimization.

Hyper Parameter Tuning Each model’s hyperparameters are tuned separately using the validation set. For batch size and initial learning rate, we search on respectively \{16, 32, 64, 128\} and \{0.01, 0.001, 0.0001\}. More detailed searching ranges are provided as follows: For the

\footnote{https://pslcdatashop.web.cmu.edu/Project?id=244}
proposed model REACT, we tune hidden size and the embedding size in \{32, 64, 128, 256, 512\}. The coefficient weight \( \rho \) is searched in \{0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 20, 30\}. For CAUSE, we search the number of basis in \{1, 2, ..., 10\}, embedding size and hidden size respectively in \{32, 64, 128, 256, 512\}, coefficient of L2 regularization in \{0, 0.001, 0.01, 0.1\}. For RPPN, we search embedding size and hidden size respectively in \{32, 64, 128, 256, 512\}. For MLE-SGLP, we search the number of Gaussian basis in \{1, 2, ..., 10\}, embedding size and hidden size respectively in \{32, 64, 128, 256, 512\}. For THP, decay ratio was searched in \{0.001, 0.005, 0.01, 0.05, 0.1\}. Gaussian basis number, group-lasso and sparse regularization are set to be respectively 10, 250 and 270 following the original paper.

6.4.4 Model Evaluation

For model evaluation, each method is evaluated from two important aspects, i.e. (1) how well it can identify the Granger causal relationship among dimensions of the multivariate point processes, (2) its ability to accurately represent the true underlying dynamics of the processes, measured by its time prediction performance.

6.4.4.1 Evaluating and Understanding Granger Causality

Evaluating Model Recovery of Granger Causality To evaluate model’s ability to identify Granger causality, we compute the Area Under the ROC Curve (AUC) of the predicted Granger causality among dimensions recovered by the model against ground truth. We further consider Kendall’s \( \tau \) coefficient and Spearman’s coefficient, which are both rank-based correlation coefficients that measures the correlation between ground truth and the recovered Granger Causality prediction. Given that the ground truth is not available in the real-world datasets, the above metrics are only considered for the evaluation of the two simulated datasets. This result is given in Table 6.1.

To better visualize the performances of all model in capturing Granger causality among dimension, the ground truth and the predictions obtained by each model in self-exciting and self-inhibiting datasets are respectively presented in Fig. 6.1 and Fig. 6.2. For this visualization, we construct a Granger causality matrix for each model from the corresponding Granger causality parameters learned during the training, namely the infectivity matrices
for MLE-SGLP and THP, the attention weights for THP, and the Granger causality graph obtained by CAUSE. Each matrix is of the size of \( D \times D \) with \( D \) representing the number of dimensions, where \( D = 10 \) in both simulated datasets. The cell in row \( d \) and column \( d' \) represents the directional Granger causality from dimension \( d' \) to dimension \( d \). If this cell is black in Fig. 6.1 and Fig. 6.2, it suggests that dimension \( d' \) does not Granger cause dimension \( d \), otherwise it means Granger causality from dimension \( d' \) to dimension \( d \) was detected.

![Table 6.1: Evaluating model ability to identify Granger causality in terms of AUC, Kendall’s \( \tau \) coefficient and Spearman coefficient in simulated datasets.](image)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Self-exciting</th>
<th></th>
<th>Self-inhibiting</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>AUC</td>
<td>Kendall</td>
<td>Spearman</td>
<td>AUC</td>
</tr>
<tr>
<td>REACT</td>
<td>0.921</td>
<td>0.406</td>
<td>0.508</td>
<td>0.843</td>
</tr>
<tr>
<td>MLE-SGLP</td>
<td>0.871</td>
<td>0.371</td>
<td>0.461</td>
<td>0.639</td>
</tr>
<tr>
<td>CAUSE</td>
<td>0.657</td>
<td>0.262</td>
<td>0.335</td>
<td>0.855</td>
</tr>
<tr>
<td>RPPN</td>
<td>0.769</td>
<td>0.269</td>
<td>0.341</td>
<td>0.811</td>
</tr>
<tr>
<td>THP</td>
<td>0.9</td>
<td>0.399</td>
<td>0.438</td>
<td>0.761</td>
</tr>
</tbody>
</table>

In the self-exciting dataset, as we can see in Fig. 6.1, the proposed model REACT (Fig. 6.1.b) recovers the Granger causality the closest to the ground truth (Fig. 6.1.a). This observation can be further confirmed in Table 6.1, where REACT achieves the highest AUC, Kendall’s \( \tau \) coefficient, and Spearman’s coefficient compared with baseline approaches. Even though it is shown in Table 6.1 that baseline THP achieves the highest result among all baseline approaches, which is comparable to REACT. Fig. 6.1 suggests that THP however was only able to recover the diagonal of the ground truth, meaning that only within-dimensional Granger causality was identified, whereas each between-dimensional Granger causality was not detected by the model. This result is equivalent to modeling each dimension independently, or in other words, modeling the \( D \)-dimensional multivariate point processes as \( D \) independent univariate point processes, which certainly fails the goal of capturing Granger causality among dimensions in multivariate point processes. Among other baseline approaches, MLE-SGLP recovers the between-dimensional Granger causality most accurately, which explains its higher AUC, Kendall’s \( \tau \) coefficient and Spearman’s coefficient compared with baselines CAUSE and RPPN. A possible explanation is that the kernel function used to generate this simulated data perfectly matches the kernel function of this model, which may result in over-fitting, further explained by its less competitive Granger
Figure 6.1: Granger causality evaluation in the simulated self-exciting dataset: Visualizations of the ground truth (a), the predicted Granger causality recovered by the proposed model REACT (b) and baseline approaches (c-f). The cell in \( d \)-th row and \( d' \)-th column being white and black respectively represents that dimension \( d' \) Granger cause and does not Granger cause dimension \( d \).

causality recovery performances in self-inhibiting dataset (later in this section), as well as time prediction power (later in Ch. 6.4.4.2).

In the task of recovering Granger causality in self-inhibiting dataset, the proposed model REACT again identifies the Granger causality in multivariate point processes more accurately compared with baseline approaches. Even though CAUSE achieves a higher AUC compared with REACT, its Kendall’s \( \tau \) coefficient and Spearman’s coefficient are not as competitive. From Fig. 6.2, a possible explanation for this observation is that CAUSE even though recovers the within-dimensional Granger causality better, it mistakenly identifies more between-dimensional Granger causality that is not actually there (white cells off diagonal misaligned with the ground truth). The combined result of Fig. 6.2 and Table 6.1
Figure 6.2: Granger causality evaluation in the simulated self-inhibiting dataset: Visualizations of the ground truth (a), the predicted Granger causality recovered by the proposed model REACT (b) and baseline approaches (c-f). The cell in \(d\)-th row and \(d'\)-th column being white and black respectively represents that dimension \(d'\) Granger cause and does not Granger cause dimension \(d\).

also shows that baseline MLE-SGLP and THP failed to identify meaningful Granger causality. A possible explanation is that both methods assume the processes to be self-exciting, therefore fail to effectively identify within or between-dimensional in this dataset where the nature of the data was reversed (i.e. self-inhibiting).

It is also worth noting that all methods model inter-arrival times assuming the influence of historical events are depend on their distances to the current time. This assumption, even though is more suitable to most real-world applications, is not applicable in the self-inhibiting dataset with its simplified parameterization which samples the history invariant to its timings (i.e. an event will regularize the intensity to the same degree regardless how far it is to the current time). This can possible explain the observation that recovering Granger causality
is more challenging in this dataset compared to the self-exciting dataset, as the information of inter-arrival times do not help the models to infer the dependencies among dimensions.

By looking at all models performance across the datasets, we see the RNN-based models REACT and RPPN can more accurately recover the true Granger causality among dimensions, which suggests the significance and benefits of representing intensity functions using RNN without restricting the parameterizations of the intensity functions as the traditional methods.

However, since RPPN does no actively learn Granger causality by having any constraints on the attention weights but only using them to provide extra quasi-causality interpretation, its performance is still not as competitive as the proposed model REACT.

To summarize, the analysis in this section shows that the proposed model REACT is able to accurately recover both within-dimensional and between-dimensional Granger causality in both simulated data, compared with all baseline approaches.

**Interpretation and Analysis of Granger Causality in User Interactive Activities**

In this section, we aim to study and understand real-world user interactive activities better based on the Granger causality inferred by the proposed model REACT. To do this, an analysis of the Granger causality matrix recovered by REACT and how to use it to explain the dynamics of user activities from each of the two real-world datasets is provided in this section.

We first analyze and study student problem-solving activities in Junyi Academy dataset. This data contains 8 domains of problems, namely algebra, analytics, arithmetic, biology, calculus, geometry, logic, and probability.

Fig. 6.3 shows the heatmap of Granger causality relationship among all types of problems estimated by REACT. First of all, we observe a strong within-dimensional Granger causality that are positive in most domains such as algebra. This observation suggests that practicing problems in each of these domains is likely to prompt subsequent practices of similar problems in the same domain. Strong dependencies between algebra and arithmetic, for example, are bidirectionally positive, implying that students may practice these two domains in a comparable fashion and that practicing one will likely encourage follow-up practices in
Geometry domain, on the contrary, demonstrates a high negative within-dimensional Granger causality, indicating that student practice intensity in this domain tends to regularize over time. One probable explanation is that, despite their importance, geometry problems are often associated with complex and abstract knowledge components such as triangles or polygons, which may be challenging for many students. As a result, students may be discouraged to tackle new similar problems but only come to practice on a relatively regular basis (i.e. negative Granger causality within the Geometry domain) and instead try other problem domains with which they are more familiar (i.e. positive Granger causality from Geometry to algebra and arithmetic).

Granger causality, on the other hand, is shown to be less significant within the dimensions of biology and logic. A possible explanation is that these problems are not popular to the targeted users of this E-learning platform, especially when the rest of the problem
domains are basic mathematics, these may not be the problems that students expected to practice. As a result, students may be less inclined to initiate follow-up practices after practicing problems in these domains.

In the IPTV dataset, the estimated Granger causality matrix is shown in Fig. 6.4. As it is shown in this figure, almost all diagonal cells are positive, suggesting that users are inclined to continue watching similar TV programs within the same categories. It is also interesting to see that kids category mostly only has positive within-dimensional Granger causality to itself. A possible explanation is that these TV programs are mostly watched by kids who are not usually interested in other programs or are restricted by their parents from watching other categories.

**Figure 6.4: Granger causality in IPTV estimated by REACT.**
On the other hand, entertainment category is shown to have strong associations with many other categories, such as from record, science, and finance. It is possible that entertainment is one of the most popular category and watching TV programs from other categories is likely to be followed by watching in entertainment. For example, after watching science, users may want to take a break by watching some entertainment TV programs which are more relaxing (i.e. science Granger causes entertainment). From the other direction, Granger causality is discovered from entertainment to mostly only the category of drama. In other words, after watching entertainment, users are less prone to watch other TV programs other than drama. Furthermore, after watching dramas, users are shown to be less inclined to watch other programs other than entertainment. Interestingly, both observations match real-world behaviors such as binge-watching [43].

6.4.4.2 Time Prediction Performance Evaluation

In this section, Root-Mean-Square Deviation (RMSE) is used to evaluate the time prediction performance of each model. This result is given in Table 6.2.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Real-world</th>
<th>Simulated</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Junyi Academy</td>
<td>IPTV</td>
<td>Self-Exciting</td>
<td>Self-Inhibiting</td>
</tr>
<tr>
<td>REACT</td>
<td>8.368</td>
<td>7.370</td>
<td>0.859</td>
<td>0.980</td>
</tr>
<tr>
<td>CAUSE</td>
<td>13.596</td>
<td>10.769</td>
<td>0.828</td>
<td>1.897</td>
</tr>
<tr>
<td>RPPN</td>
<td>8.915</td>
<td>9.860</td>
<td>0.909</td>
<td>1.037</td>
</tr>
<tr>
<td>MLE-SGLP</td>
<td>14.140</td>
<td>11.925</td>
<td>0.963</td>
<td>1.636</td>
</tr>
<tr>
<td>THP</td>
<td>13.856</td>
<td>10.097</td>
<td>1.353</td>
<td>1.497</td>
</tr>
</tbody>
</table>

Table 6.2: Time prediction comparison of all models in terms of RMSE.

As it is shown in this table, REACT usually achieves a better or comparable time prediction than baseline approaches across datasets, which suggests is ability to accurately represent the underlying dynamic of the different data. It is also observed that the results in simulated self-exciting data for all models are all generally good. One probable explanation is that, due to the design of the intensity functions, all models can manage the modeling of this dataset’s self-exciting nature. However, the performance of baselines such as CAUSE, THP and MLE-SGLP are less ideal in other datasets. Note that the intensity functions of these models are all inspired by Hawkes processes that are designed for self-exciting data. And this assumption is violated in self-inhibiting data and the possibly is inconsistent with the more
complicated real-world datasets, which may explain their less ideal prediction performance. Indeed, by looking at the performance across all datasets, only REACT and RNNP consistently achieve better time prediction performances. This observation again highlights the limitation of certain point process models where the kernel function is simplified to model only restricted number of scenarios such as self-exciting. On the other hand, RNN-based methods REACT and RPPN have more flexible representations of the intensity function, which may explain their consistently better performances across different datasets compared with CAUSE, MLE-SGLP, THP. However, unlike REACT that models each dimension using one LSTM, RPPN uses only one LSTM and its hidden state to characterize the history of all dimensions. This method compromises the representation of multivariate point processes if not the dynamics of all dimensions are all similar, and may explain RPPN’s less competitive time prediction performance compared with REACT especially in real-world datasets where the data dynamics may be more complicated.

To summarize, this section shows that the REACT consistently outperforms baseline approaches, for its ability to accurately characterize complicate and different dynamics in multivariate point processes.

6.4.4.3 Ablation Study

To show the importance of modeling Granger causality in multivariate point processes, we conduct an ablation study in this section.

Specifically, we evaluate REACT’s performance in recovering truth Granger causality among dimensions (i.e. AUC, Kendall’s $\tau$ coefficient, and Spearman’s rank correlation coefficient in simulated datasets), and its effectiveness in modeling activity dynamic (i.e. time prediction RMSE in all datasets). The following regularization coefficient is used $\{0, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 20, 30\}$. Note that having $\rho = 0$ represents that no Granger causality constraint is applied. The line plots of these evaluation metrics (y-axis) with respect to different choices of $\rho$’s (x-axis) for all dataset are provided in Fig. 6.5. Note that increasing $\rho$ is the same to increase model’s sensitivity to Granger non-causality among dimensions, or in other words, decreasing model’s sensitivity to noises in detecting causality among dimensions.

It is shown in this figure that in all datasets, regularizing between-dimensional depen-
dependencies by using some $\rho > 0$ in general improves model’s ability to recover true Granger causality among dimensions, as well as to improve its time prediction performance, compared it its variant where $\rho = 0$. This observation suggests the importance of controlling model’s sensitivity of Granger causality among dimensions by reinforcing the group sparsity of input gate weight $W_d$ for all dimension $d$. Particularly, time prediction performance in terms of RMSE is shown to first increase and then drop with the increase of $\rho$. And this trend is usually in accordance to the trend of AUC, Kenall’s $\tau$ and Spearman coefficients. An explanation for this observation is that, when $\rho$ is small, model is usually very sensitive of noises and mistaken these noises as between-dimensional dependencies (suggested by low AUC, Kenall’s $\tau$ and Spearman coefficient), failing to truly capture the underlying dynamics between dimensions, and leading to less ideal time prediction performance (suggested by high RMSE). However, as $\rho$ increases, model’s sensitivity to noises also decreases as the

Figure 6.5: Ablation study result of REACT.
model is more and more close to find the ground truth. But when $\rho$ is too high, the model is inclined to believe there is little or no Granger causality among dimensions. In other words, true between-dimensional dependencies are regarded as noises, leading to worse performance in both time and Granger causality predictions.

6.5 Conclusion

In this chapter, we propose a novel approach REACT that models Granger causality in multivariate point processes via RNNs. Specifically, we proposed to model each dimension via one LSTM and propose an input format that can keep track of the dependencies among LSTMs so that the relation between each pair of dimensions can be maintained. We are also the first that gave a definition of Granger causality in neural multivariate point processes that does not rely on a specific format of intensity function, and that can be easily extended to a more general framework of neural point process models.

Our experiments in two simulated datasets show that our model is able to more accurately recover the Granger causality among dimensions compared to baseline approaches in terms of AUC, Kendall’s $\tau$ coefficient and Spearman’s rank correlation coefficient. Furthermore, our analysis in the two real-world datasets further demonstrates our model’s power in capturing meaningful and insightful relation among different types of real-world user activities.

Finally, an ablation study demonstrates the significance of our proposed framework in modeling Granger causality, where we discovered that having a relative stronger regularization on Granger causality helps in decreasing model sensitivity to noises and achieving better Granger causality recover and data dynamic representation.
CHAPTER 7

Conclusion

In this dissertation, we explored and studied four research questions centering around the limitations in the literature of temporal point processes. To answer these research questions, we proposed four interpretable and high-capacity temporal point process models, i.e. Relaxed Clustered Hawkes process with a Gamma prior (RCHawkes-Gamma) that captures the underlying data group structure (RQ1), Stimuli-Sensitive Hawkes Process model (SSHP) that effectively represents external stimuli (RQ2), Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt) that captures the bidirectional relations between markers and activity dynamics while revealing detailed temporal activity patterns (RQ3), and REcurrent Granger Causality in pointT process (REACT) that models Granger causality in multivariate point processes even with nonlinear and complex data dynamics. In particular, by answering four important research questions, we made the following contributions:

- **Modeling cluster structure in Hawkes processes (Ch. 3).** To answer RQ1, that is how to model the relationship among point processes without assuming that they are i.i.d, we proposed a novel Relaxed Clustered Hawkes process with a Gamma prior (RCHawkes-Gamma) that models all point processes jointly while automatically capturing a cluster structure of the data by learning a low-dimensional representation of a matrix format of the processes. Evaluating RCHawkes-Gamma on simulated and real-world user activity data, our extensive experiments demonstrated the superior predictive power of our model in recovering the group structure of the data, and in accurately predicting user future activity times. Our analysis also showed that our model is able to identify meaningful clusters of user activity dynamic, providing useful interpretations such as user interaction patterns and preferences.

- **Modeling external stimuli in Hawkes processes (Ch. 4).** To answer RQ2, i.e., how to effectively represent external stimuli in temporal point process modeling, we proposed our Stimuli-Sensitive Hawkes Process (SSHP) model, which is designed to capture important types of external stimuli. Our experiments demonstrated our
model’s ability to successfully recover the patterns of user activity reacting to three external stimuli over time, and illustrated its better time prediction performance compared with baseline approaches. And in our analysis, our model is shown to identify several meaningful user groups which helps understand users activity dynamics by analyzing how they react to these stimuli.

• **Modeling marked temporal point processes (Ch. 5).** To answer RQ3, i.e. how to efficiently model markers in TPPs while capturing the interrelationship between markers and activity timings, we proposed Marked Point Processes via Memory-Enhanced Neural Networks (MoMENt) that models fine-grained and interpretable bi-directional influences between activity dynamics and activity markers. Using several benchmark real-world user activity datasets, our experiments demonstrated the effectiveness of the proposed model MoMENt in predicting future activity time, type and marker. Our case study also revealed interpretable representations of user-item relation, e.g. students knowledge over practicing over time, and shed light on how users’ future behaviors are influenced by their past experiences from both viewpoints of markers and activity dynamics characterized by time and type.

• **Modeling Granger causality in temporal point processes (Ch. 6).** To answer RQ4, i.e. how to efficiently capture Granger causality in temporal point processes, we proposed REcurrent GrAnger Causality in pointT process (REACT) that models Granger causality in multivariate point processes. Specifically, we proposed to model each dimension via one LSTM and proposed an input format that keeps track of the dependencies among LSTMs so that the relation among dimensions can be well-maintained. We are also the first to give the formulation of Granger causality in point processes that can be easily extended to a more general framework of neural TPPs for the modeling of complex data dynamics. Our experiments and analyses on several real-world user activity datasets demonstrated the power of REACT in accurately predicting users’ future activity timings while successfully recovering Granger causality within and between dimensions which provides meaningful hints of user preferences through their interaction patterns.

• **Solving problems in education and recommendation domains using TPPs.** Considering these domains’ importance, the limitations of the state-of-the-arts in these
domains, and the fitness of our problems to these domains, we applied our models to the domains of education and recommendation. By addressing the limitations of the literature of TPPs as well as the state-of-the-arts designed for these domains, our models are shown to achieve better predictions while introducing meaningful interpretations. Our models provide feasible solutions to important questions, particularly the modeling and the understanding of student procrastination in online courses, student knowledge tracing in online tutoring platforms, as well as the modeling and analysis of user interactions, such as users’ digital footprints, their online ratings, and their TV-watching behaviors. These solutions can help achieve important goals in education and recommendation domains, such as improving teaching and learning quality in online education, as well as understanding and improving user behavior and satisfaction.

7.1 Limitations

In this dissertation, following the convention in the TPPs literature, we learned point processes by modeling all activities during an observation window. However, this method is related to the edge effect, which is a limitation of temporal point process models in the literature, that is the possible effects of unobserved activities that exert on the activities in the observation windows that are not captured. A possible solution to alleviate this effect is to learn all activities in the sequence. However, this method could be computational expensive and even infeasible, especially in applications when there exist extremely long sequences. Therefore, in the future, it is beneficial to develop techniques such as parallel computing or distributed learning to alleviate the edge effect and the scalability concern in point process modeling.

As another limitation, in Ch. 4, we propose a stimuli-sensitive Hawkes process model that is shown to effectively represents external stimuli in online courses exert on students considering the task of modeling student procrastination. Particularly, we consider the effects of assignment deadline, assignment opening, and student habits in completing online course assignments. However, in other applications such as user checking-in behavior modeling, the distribution of user-item interactions may require different designs of parameterizations to more accurately represent external stimuli.

From the viewpoint of applications, to model procrastination, we use delay measure as
a proxy which is computed as the normalized difference between a student’s first learning activity to the assignment open time. Self-reported procrastination measures, on the other hand, could have helped in labeling delaying behaviors more accurately as procrastination. However, due to reasons such as privacy concerns, publicly available large-scaled student data containing such information is exceedingly scarce, making the procrastination modeling with self-reported measures impracticable. As a result, it may be advantageous in the future to research and develop more procrastination measures with the help of self-reported data that accurately reflect procrastinating behaviors, and then revisit the analyses provided by this dissertation to gain a more comprehensive understanding of student procrastination.

7.2 Future Work

From both viewpoints of methodology and application discussed in this dissertation, we want to point out some exciting directions for future work as follows:

- **Spatial-temporal point process extensions.** Spatial-temporal point processes, which can be thought of as a more advanced variant of temporal point processes, consider not only the temporal dynamics of the processes, but also the spatial aspect of the data. By introducing penalties for learning the spatial part of the data, our models can be easily extended to spatial-temporal point processes. As a result, the extended versions of our models can help with a wider range of real-world problems involving spatial information, such as earthquake and aftershock modeling, urban mobility, and disease diffusing modeling.

- **Wassestein temporal point process models.** In this dissertation, we provide frameworks to model temporal point processes based on maximum likelihood estimation for the learning, that is asymptotically equivalent to the minimization of Kullback–Leibler divergence. However, performing log-likelihood estimation could sometimes result in the estimate of local maximum due to the possible non-convex nature of complex log-likelihood function. Wassestein-distance on the other hand, can be used as an alternative to point process modeling, as it has been found to be less sensitive to outliers. As a result, one promising future approach is to extend our frameworks to construct likelihood-free models utilizing Wasserstein-distance-based methods.
Temporal point process models for interventions. In this dissertation, our focus has been giving to the modeling of user activity dynamics using temporal point process models. Beyond the modeling of dynamics, one exciting direction is to explore other potentials of point process models such as interventions, by introducing sequences to mitigate or influence the dynamic towards preferred ones, e.g., by maximizing the expected reward function using reinforcement learning. For a more concrete example, in student modeling, in addition to modeling student learning dynamics, one potential application is to nudge students to learn at the appropriate moments to possibly reduce their procrastination.
BIBLIOGRAPHY


