Temporal Cross-Effects in Knowledge Tracing

Chenyang Wang[1], Weizhi Ma[1], Min Zhang[1]*, Chuancheng Lv[1], Fengyuan Wan[1], Huijie Lin[2],
Taoran Tang[2], Yiqun Liu[1], Shaoping Ma[1]
[1]Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing, China
[2]Netease Youdao, Beijing, China
wangcy18@mails.tsinghua.edu.cn, z-m@tsinghua.edu.cn

ABSTRACT

Knowledge tracing (KT) aims to model students' knowledge level based on their historical performance, which plays an important role in computer-assisted education and adaptive learning. Recent studies try to take temporal effects of past interactions into consideration, such as the forgetting behavior. However, existing work mainly relies on time-related features or a global decay function to model the time-sensitive effects. Fine-grained temporal dynamics of different cross-skill impacts have not been well studied (named as *temporal cross-effects*). For example, cross-effects on some difficult skills may drop quickly, and the effects caused by distinct previous interactions may also have different temporal evolutions, which cannot be captured in a global way.

In this work, we investigate fine-grained temporal cross-effects between different skills in KT. We first validate the existence of temporal cross-effects in real-world datasets through empirical studies. Then, a novel model, *HawkesKT*, is proposed to explicitly model the temporal cross-effects inspired by the point process, where each previous interaction will have different time-sensitive impacts on the mastery of the target skill. HawkesKT adopts two components to model temporal cross-effects: 1) **mutual excitation** represents the degree of cross-effects and 2) **kernel function** controls the adaptive temporal evolution. To the best of our knowledge, we are the first to introduce Hawkes process to model temporal cross-effects in KT. Extensive experiments on three benchmark datasets show that HawkesKT is superior to state-of-the-art KT methods. Remarkably, our method also exhibits excellent interpretability and shows significant advantages in training efficiency, which makes it more applicable in real-world large-scale educational settings.

CCS CONCEPTS

• **Social and professional topics** → *Student assessment*; • **Applied computing** → **Learning management systems**.

KEYWORDS

Educational data mining, Knowledge tracing, Temporal cross-effects, Hawkes process, Collaborative filtering

WSDM '21, March 8–12, 2021, Virtual Event, Israel
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441802

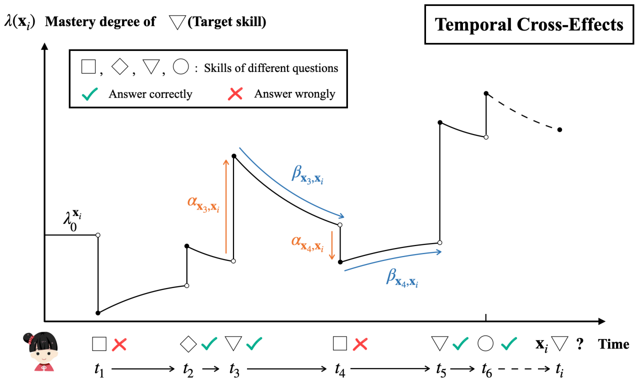

$\lambda(\mathbf{x}_i)$ Mastery degree of ▽(Target skill)
Temporal Cross-Effects
□, ◇, ▽, ○ : Skills of different questions
✓ Answer correctly
✗ Answer wrongly
$\lambda_0^{\mathbf{x}_i}$
$\alpha_{\mathbf{x}_3,\mathbf{x}_i}$
$\beta_{\mathbf{x}_3,\mathbf{x}_i}$
$\alpha_{\mathbf{x}_4,\mathbf{x}_i}$
$\beta_{\mathbf{x}_4,\mathbf{x}_i}$
□✗ ◇✓ ▽✓ □✗ ▽✓ ○✓ $\mathbf{x}_i$▽ ?
Time
t_1 t_2 t_3 t_4 t_5 t_6 t_i

Figure 1: Illustration of how the mastery degree of the target skill (▽) is influenced by temporal cross-effects. The base knowledge level of the target skill is denoted as $\lambda_0^{\mathbf{x}_i}$, and each past interaction will have adaptive impact ($\alpha_{\mathbf{x}_j,\mathbf{x}_i}$) on target skill's mastery degree. Furthermore, all the effects evolve differently with time ($\beta_{\mathbf{x}_j,\mathbf{x}_i}$), depending on both previous interactions and the target skill.

ACM Reference Format:
Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, Shaoping Ma. 2021. Temporal Cross-Effects in Knowledge Tracing. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441802

1 INTRODUCTION

Nowadays, computer-assisted learning (CAL) has been a vital part of education methodologies. It is increasingly accessible for students to study on all kinds of intelligent tutoring platforms. Besides, the abundant learning logs in CAL systems enable them to provide personalized learning trajectories by analyzing data from students' learning history. Skills that are too difficult or have already been mastered can be identified and only the most suitable learning materials will be presented [31].

A key problem in learner data analysis is the assessment of student's knowledge state. Knowledge tracing (KT) is such a task of predicting students' future performance (responses to assessment questions) given their past interactions in educational applications [6]. It is challenging since many factors are involved in the

*Corresponding author.

learning process, such as one's ability to acquire knowledge, temporal dynamics, and human cognition [29]. Some traditional methods use the Hidden Markov Model to capture how a student's knowledge evolves, among which the most popular method is Bayesian Knowledge Tracing (BKT) [6, 15]. Another line of work centers around item response theory (IRT), which aims to learn common factors to generalize observations [3, 16, 27]. Recently, with the rapid progress in deep learning, some RNN-based methods [24, 29] are proposed to model long dependencies between interactions.

In this study, we want to address that learning is a dynamic process and there exist *temporal cross-effects* in KT. For one thing, the mastery of a skill is not only influenced by previous interactions of the same skill, but also the others (cross-effects). For another, the temporal evolution for different cross-skill effects can also be different. As shown in Figure 1, each previous interaction will take different immediate effects on the target skill. Besides, although such effects all decay with time, their decay rates differ from each other, which we call temporal cross-effects in this paper. Some skills may be too easy to forget, and the effects caused by different previous interactions may also have different temporal evolutions.

There are a few recent studies beginning to partially address the above temporal factors in KT [11, 14, 24, 30, 37]. These methods mainly focus on discretizing time into slots or extracting handcrafted features. Some researches move one step forward to use a global decay function to control the forgetting behavior [11, 14]. However, as shown above, learning is an adaptive and dynamic process. Each previous interaction will take effect with different temporal dynamics. The temporal cross-effects in KT may depend on both previous interactions and the target skill, which cannot be fully captured in a global way.

In this paper, we first validate the existence of temporal cross-effects in KT through empirical studies. Based on the analyses of mutual information between student interaction pairs in real-world datasets, we find the temporal evolution is indeed distinct for different cross-skill effects. Then, we introduce point process to adaptively model temporal cross-effects in KT. A novel model, HawkesKT, is proposed inspired by Hawkes process, which is a variant of point process utilizing intensity function to model mutual excitation between events localized in time. In KT scenario, the basic event relies on the skill and response of each interaction. Specifically, to predict one's knowledge state of the target skill, both the accumulative effects of historical interactions and their evolutions over time are naturally characterized by the designed intensity function. Besides, such cross-effects and temporal evolutions are unique for different historical interactions and target skills. Collaborative filtering is also utilized here to reduce the high complexity of calculating all skill pairs' parameters. Different from deep learning based state-of-the-art methods, the parameters in our model are highly interpretable, which can be used to automatically discover latent relationships between skills. Actually, HawkesKT reveals a brand new branch of approaches for KT, which is different from various existing methods. The main contributions of this work can be summarized as follows:

- We present *temporal cross-effects* in KT through empirical studies on real-world datasets. The fine-grained temporal evolution of different cross-skill effects should be taken into consideration.
- We propose a novel model HawkesKT to address temporal cross-effects in KT based on point process. To the best of our knowledge, we are the first to introduce Hawkes process into this field.
- Comparative experiments on three real-world datasets show the effectiveness of HawkesKT. Our method also exhibits great interpretability and has significant advantages in training efficiency.

2 RELATED WORK

2.1 Knowledge Tracing

Classically, there are two lines of work about KT. Some studies are based on the Hidden Markov Model. The most representative method is BKT [6], which describes a student's knowledge state with a binary variable. The other line of work is based on factor analysis. IRT [12] tends to assume and modify the potential traits of subjects on the basis of observing test responses. AFM [3] and PFA [27] are logistic regression models that predict the performance based on different previous information. Further, the recently proposed KTM [34] leverages Factorization Machines to model pairwise interactions among features and is shown to encompass all above factor analysis models.

As deep learning is making rapid progress in a range of domains, RNN is utilized to capture complex dependencies between interactions. DKT [29] uses the hidden state of RNN at each step to represent student's knowledge state and gets promising results generally. Subsequently, many studies follow DKT to extend its capacity [5, 17, 32], and some work tries to explore other model structures (e.g. memory network, self-attention) to get higher expressiveness [2, 25, 39]. However, all the above methods neglect the importance of temporal information. As a result, given an interaction sequence, they cannot exactly estimate a student's changing knowledge state at different times.

2.2 Temporal Dynamics in Knowledge Tracing

Typically, there exists a lot of temporal information in KT, and the impact of temporal dynamics on predicting future response has gradually emerged. Many studies focus on the forgetting behavior in the learning process. Early explorations mainly incorporate a lag time factor into BKT or PFA [28, 30]. DKT-t [20] and DKT-Forgetting [24] introduce different time-based features into DKT. DKT-Forgetting considers repeated and sequence time gap, as well as the number of past trials, which is a state-of-the-art method with temporal information. More recently, some work utilizes a decay function to control the forgetting behavior [11, 14], which assumes interactions happening recently have larger impacts.

However, these studies either rely on hand-crafted features or model the continuously changing effects with a global decay rate. Differently, our HawkesKT explicitly models temporal cross-effects of each previous interaction, which can capture distinct temporal trends of different cross-skill effects.

2.3 Hawkes Process

Point process is known to be good at modeling sequential events localized in time [7]. There has been a lot of applications of point process, including earthquakes prediction [21], user influence in social network [33, 40] and paper citation count [38]. Among variants of point process, Hawkes process [13] explicitly models self-exciting

Table 1: Top-10 frequency skills in ASSISTments 12-13.

ID	Skill Name
0	Equation Solving Two or Fewer Steps
1	Addition and Subtraction Integers
2	Addition and Subtraction Fractions
3	Conversion of Fraction Decimals Percents
4	Multiplication and Division Integers
5	Multiplication and Division Positive Decimals
6	Order of Operations All
7	Multiplication Fractions
8	Division Fractions
9	Equation Solving More Than Two Steps

and mutual-exciting characteristics of sequential events, and corresponding temporal trends are controlled by kernel function in the intensity function. Recently, Hawkes process has been gaining increasingly attraction and shows great effectiveness in various domains, such as online activity prediction and personalized recommendation [8, 22, 35].

3 EMPIRICAL STUDY

In this section, we first formally define the knowledge tracing task and introduce the symbols used in this paper. Then we validate whether there exist temporal cross-effects in real-world educational datasets.

3.1 Knowledge Tracing Task Setup

Definition 3.1 (Knowledge Tracing Task). **Given** a student's interaction sequence $S_t = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_n\}$, knowledge tracing (KT) aims at predicting whether he/she can answer the question correctly in the next interaction $\mathbf{x}_{n+1}$.

In this study, an interaction $\mathbf{x}_i$ is defined as a tuple (q_i, t_i, a_i), including the question q_i that the student attempts to answer at timestamp t_i, and corresponding response $a_i \in \{0, 1\}$ (correctness of the answer, 1 means right). The sequence is sorted by time in an ascending order, i.e. $t_i < t_j$ for any $i < j$. Besides, to identify the skill involved in each question, we have a mapping $s(\cdot)$ from questions to skills, which can get the corresponding skill id $s(q_i)$ of question q_i. Here a_{n+1} is the target to predict given S_t and (q_{n+1}, t_{n+1}).

3.2 Dataset Description

Here we use a real-world benchmark dataset, ASSISTments 12-13, to conduct empirical studies. ASSISTments is an online tutoring system that teaches and assesses students in mathematics, and this series of datasets is often utilized in related research [10]. There are totally 2.7M interactions in the dataset, involving 265 skills. More detailed information can be found in Section 5.1.1.

To facilitate the understanding, we will mainly demonstrate the analyses results about the top-10 skills with the highest frequency. Table 1 shows the name of these skills in the dataset. Note that these skill ids will be used throughout the paper.

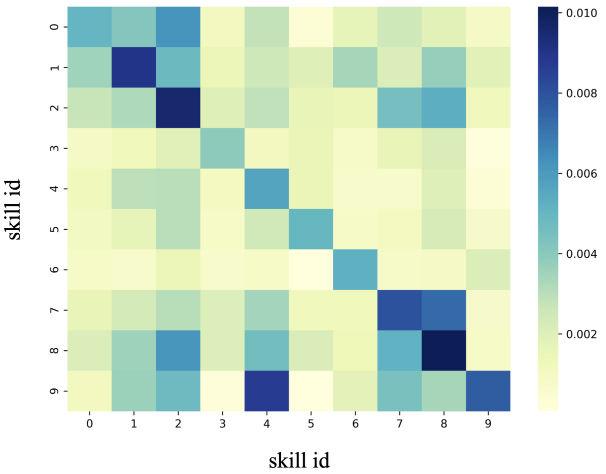


Figure 2: CMI for all the skill pairs (cross-effects). The y-axis is the skill id of the pre-interaction, and the x-axis is the skill id of the post-interaction.

3.3 Temporal Cross-Effects

To validate whether there exist temporal cross-effects, we first define the *conditional mutual information* (CMI) between student interaction pairs, which will be used in the following analyses.

Definition 3.2 (Conditional Mutual Information). **Given** a restrictive condition c, we can find all the interaction pairs $(\mathbf{x}_i, \mathbf{x}_j)$ that cater to c in each student's interaction sequence. If we view the response of pre- and post-interaction (a_i and a_j) as a random variable respectively, the conditional mutual information is defined as

$$CMI(a_i; a_j) = \sum_{a_i \in \{0,1\}} \sum_{a_j \in \{0,1\}} P(a_i, a_j) \cdot \log \frac{P(a_i, a_j)}{P(a_i)P(a_j)}. \quad (1)$$

Here the condition c can be specific skills of the pre- and post-interaction, or the time interval between the two interactions. And the probabilities in the definition can be derived by counting frequencies within all the satisfied interaction pairs. Actually, CMI reflects the degree of dependency between pre- and post-interaction under the restrictive condition.

First, we restrict the skill of pre- and post-interaction to validate the cross-effects between skills. Note that if two skills are totally independent, the corresponding CMI should be 0. Figure 2 shows the CMI of all the combinations of skill pairs. The y-axis is the skill id of the pre-interaction, and the x-axis is the skill id of the post-interaction. We can see the effects are generally the largest for interactions of the same skill (diagonal). However, there exist obvious cross-effects between different skills, such as the condition when the pre- and post-interaction is 9 and 4, respectively. And the dependencies are high within the skill group {0, 1, 2} and {7, 8, 9}, which makes sense because these skills are generally perceived to be related. As a result, when predicting the mastery degree of the target skill, it is important to focus on not only previous interactions with the same skill, but also other related ones.

Second, we move forward to investigate the temporal evolution of different cross-effects. In addition to the restriction on the skill

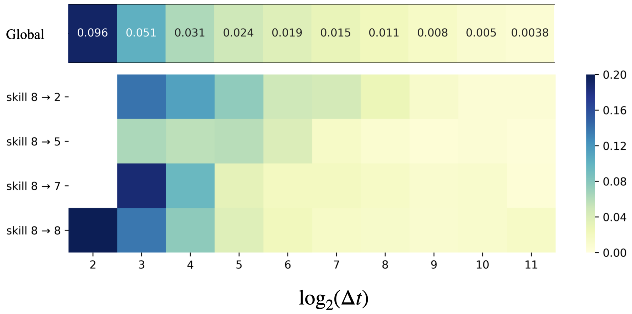


Figure 3: The temporal evolution of CMI for the global trend and some representative skill pairs (temporal cross-effects).

of pre- and post-interaction like before, we further group the interaction pairs according to the log time interval between the two interactions. Figure 3 shows the global trend and some representative skill pairs. The time interval after log transformation starts from 2 and some grids are masked because there are not enough interaction pairs (less than 50) under those conditions. Globally speaking, the overall temporal evolution exhibits a decaying form because of the forgetting behavior, which is consistent with previous studies [24]. However, it is noteworthy that for different skill pairs, the decay rates differ from each other obviously, which we call temporal cross-effects. For example, when the pre- and post-interaction is 8 and 7 respectively, the CMI in the short term is large because they are highly related, but it decays quickly with time. On the other hand, the CMI between 8 and 2 is smaller and decays slower, which is reasonable because skill 2 is relatively easier and these two skills are not directly correlated. Note that there are also many other skill pairs demonstrating significant differences w.r.t. the temporal evolution, in which case a global decay function in previous work is not sufficient.

Therefore, to capture such temporal cross-effects shown in the above empirical studies, it is important to model fine-grained forgetting behavior in KT, where both the cross-skill effects and adaptive decay rates should be taken into consideration.

4 METHODOLOGY

4.1 Preliminaries about Hawkes Process

Formally, a temporal point process is a random process of which the realization consists of a list of discrete events localized in time, $\{t_n\}_{n \in \mathbb{N}}$ with the time $t_n \in \mathbb{R}^+$. In the KT scenario, it represents a series of timestamps when a student answers distinct questions correctly/incorrectly, which constitute the basic events in temporal point process. Given the history time of past events S_t, temporal point process introduces conditional intensity function $\lambda(t|S_t)$, representing a stochastic model for the time of the next event. For simplicity, we omit the conditional sign as $\lambda(t)$ in the following parts. Then, the probability for the occurrence of a new event within a small time window $[t, t + dt)$ can be expressed as [1, 35]:

$$\lambda(t)\ dt = \mathbb{P}\{\text{event in } [t, t + dt) \mid S_t\}\ . \tag{2}$$

As for the concrete form of the intensity function $\lambda(t)$, various models differ from each other. As a popular and powerful variant, Hawkes process models the excitation between events, whose intensity function takes the form of:

$$\lambda(t) = \lambda_0 + \alpha \sum_{t_j < t} \kappa(t - t_j)\ , \tag{3}$$

where λ_0 is the base intensity and every history event has an addictive effect α. The effects vary with the time gap and the triggering kernel $\kappa(\cdot)$ controls corresponding temporal characteristics.

4.2 HawkesKT Model

Inspired by the intensity function in Hawkes process, we design $\lambda(\mathbf{x}_i)$ to represent how likely a student will answer the question q_i correctly at t_i given the history interactions S_{t_i}. To model the temporal cross-effects in KT, mutual excitation $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ is used to capture cross-skill effects, and the fine-grained temporal evolution is addressed in the designed kernel function $\kappa_{\mathbf{x}_j,\mathbf{x}_i}(\cdot)$, leading to the intensity function in the following form:

$$\lambda(\mathbf{x}_i) = \overbrace{\lambda_0^{\mathbf{x}_i}}^{\text{base}} + \overbrace{\sum_{\mathbf{x}_j \in S_{t_i}} \alpha_{\mathbf{x}_j,\mathbf{x}_i} \cdot \kappa_{\mathbf{x}_j,\mathbf{x}_i}(t_i - t_j)}^{\text{temporal cross-effects}}\ . \tag{4}$$

Here the total intensity is composed of the base intensity $\lambda_0^{\mathbf{x}_i}$ and the temporal cross-effects part. Base intensity aims to capture the difficulty of the target question itself, while temporal cross-effects model the adaptive time-varying impacts of previous interactions.

4.2.1 *Base Intensity*.

We notice that previous studies usually do not take question index into consideration, probably due to data sparsity and the large number of parameters it brings. In deep-learning-based models, it is too expensive to give each question a separate embedding. As a result, using skills to index questions is an effective way to avoid overfitting and overparameterization. However, in practice, distinct questions have different levels of difficulties, even with the same skill. Lacking the modeling of questions will lead to less expressiveness and flexibility.

Here we leverage base intensity to capture the difficulty degree of both skills and questions, which is defined as follows:

$$\lambda_0^{\mathbf{x}_i} = \lambda_0^{q_i} + \lambda_0^{s(q_i)}\ , \tag{5}$$

where $\lambda_0^{q_i}$ and $\lambda_0^{s(q_i)}$ are parameters for each question and skill, respectively. When predicting the response of the target interaction $\mathbf{x}_i$, every previous interaction will take effects on the basis of this base intensity $\lambda_0^{\mathbf{x}_i}$, which represent inherent characteristics of the target interaction. In this way, only one parameter is introduced for each question, which strikes the balance between modeling individual question and avoiding overparameterization.

4.2.2 *Temporal Cross-Effects*.

As shown in Section 3, previous events have different impacts on the target interaction, and the effects decay as time goes by. Moreover, the decay rates differ from each other, which are related to both historical interactions and the target skill. In this part, we focus to model such adaptive temporal cross-effects in KT. There are mainly two components here: (1) **mutual excitation** $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ that controls the degree of immediate effects, and (2) **kernel function** $\kappa_{\mathbf{x}_j,\mathbf{x}_i}(t_i - t_j)$ that controls fine-grained temporal dynamics of cross-effects.

First, we use $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ to model to what extent the previous interaction $\mathbf{x}_j$ will influence the response in the target interaction $\mathbf{x}_i$.

Here we view the skill-response pair $(s(q_j), a_j)$ as a basic event in the history sequence, and the skill index $s(q_i)$ is the target will be influenced[2]. In this way, assuming there are $|\mathcal{S}|$ skills in total, the mutual excitation $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ can be resolved as a parameter matrix with shape $2|\mathcal{S}| \times |\mathcal{S}|$. The first dimension represents the status of the history interaction, and the second dimension stands for the target skill to predict. It is noteworthy that the mutual excitation $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ inherently encompasses the relationship between each skill pair.

Second, to model the forgetting behavior, we chose to use the exponential function as the kernel function:

$$\kappa_{\mathbf{x}_j,\mathbf{x}_i}(t_i - t_j) = \exp\left(-(1 + \beta_{\mathbf{x}_j,\mathbf{x}_i}) \log(t_i - t_j)\right) , \tag{6}$$

where $\beta_{\mathbf{x}_j,\mathbf{x}_i}$ is another core parameter controlling fine-grained decay rates under different circumstances. Specifically, given the target skill, the effects of historical events with different skills and responses will have adaptive decay rates. As for the form of the kernel function, the exponential function is a natural choice for approximating the forgetting curve. It is also commonly used in many applications of Hawkes process and is proved efficient most of the time [9, 23, 38]. Besides, we find applying log transformation to the time interval $t_i - t_j$ is important because the time interval often demonstrates a long-tail distribution. And the exponential function actually turns to a power function $1/(t_i - t_j)^{1+\beta_{\mathbf{x}_j,\mathbf{x}_i}}$ under this setting. One can also design other function forms to fit different real-world application scenarios.

Subsequently, with the intensity value $\lambda(\mathbf{x}_i)$, the probability of answering the question correctly in interaction $\mathbf{x}_i$ is predicted by applying the sigmoid function to the intensity value:

$$\hat{y}_i := P(a_i = 1) = \frac{1}{1 + \exp(-\lambda(\mathbf{x}_i))} . \tag{7}$$

4.3 Reparameterization Method

Next, we focus on how to deal with the parameters in our model. Besides base intensity, the core parameters are $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ and $\beta_{\mathbf{x}_j,\mathbf{x}_i}$. Generally, they are modeled as a matrix in Hawkes process respectively, where each entry represents the parameter for a specific combination of history and target event:

$$\mathbf{A} \in \mathbb{R}^{2|\mathcal{S}|\times|\mathcal{S}|}, \mathbf{B} \in \mathbb{R}^{2|\mathcal{S}|\times|\mathcal{S}|} , \tag{8}$$

The first dimension stands for the skill-response pair $(s(q_j), a_j)$ and the second dimension indexes the target skill $s(q_i)$ to predict.

Although directly optimizing the parameter matrix is an intuitive solution, there are two major problems. First, the event pairs existing in the dataset are usually sparse compared to the total $2|\mathcal{S}|^2$ kinds of combinations. As a result, only a few parameters will get updated, and the number of parameters will be huge if $|\mathcal{S}|$ is large. Second, the parameters for different pairs are independent, and hence the patterns of temporal cross-effects learned from data cannot propagate as well as generalize to unseen cases. Therefore, we introduce matrix factorization as a reparameterization method to take advantage of collaborative filtering [19] and reduce the total number of parameters, which is often utilized in recommender systems [4, 36].

Collaborative filtering assumes similar history events have similar effects on the target interaction. We can encode skill-response pair and the target skill into the same vector space, and use inner product to derive the parameters for each combination. In this way, we will have two factor matrices for each set of core parameters:

$$\mathbf{P}_A \in \mathbb{R}^{2|\mathcal{S}|\times D}, \mathbf{Q}_A \in \mathbb{R}^{|\mathcal{S}|\times D},$$
$$\mathbf{P}_B \in \mathbb{R}^{2|\mathcal{S}|\times D}, \mathbf{Q}_B \in \mathbb{R}^{|\mathcal{S}|\times D}.$$

Here D denotes the dimension of the hidden space. Then the specific $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ and $\beta_{\mathbf{x}_j,\mathbf{x}_i}$ can be calculated as:

$$\alpha_{\mathbf{x}_j,\mathbf{x}_i} = \sum_{d=1}^{D} p_A^{s(q_j)+a_j|\mathcal{S}|} \cdot q_A^{s(q_i)} , \tag{9}$$

$$\beta_{\mathbf{x}_j,\mathbf{x}_i} = \sum_{d=1}^{D} p_B^{s(q_j)+a_j|\mathcal{S}|} \cdot q_B^{s(q_i)} . \tag{10}$$

In this way, the number of parameters will be reduced from $O(4|\mathcal{S}|^2)$ to $O(6|\mathcal{S}|D)$ considering $D \ll |\mathcal{S}|$. Besides, benefiting from collaborative filtering, the learned patterns of temporal cross-effects are encoded in embeddings of each dimension. This will be extremely helpful to model the temporal cross-effects of rare interaction pairs and understand latent relationships between skills.

4.4 Parameter Learning

In summary, the parameters in HawkesKT are base intensity $\lambda_0^{q_i}$, $\lambda_0^{s(q_i)}$, and factor matrices $\{\mathbf{P}_A, \mathbf{Q}_A, \mathbf{P}_B, \mathbf{Q}_B\}$. To jointly learn these parameters, we optimize a standard cross-entropy loss between the predicted probability $\hat{y}_{n+1}$ and the true response a_{n+1}:

$$\mathcal{L} = -\sum_{n} \left(a_{n+1} \log \hat{y}_{n+1} + (1 - a_{n+1}) \log(1 - \hat{y}_{n+1})\right) \tag{11}$$

Due to the success of Adam algorithm [18], we use Adam as the learning algorithm. We also add weight decay on factor matrices.

4.5 Prerequisite Score

Besides predicting future performance, HawkesKT is also able to automatically discover latent skill relationships based on meaningful parameters. Note that the parameter $\alpha_{\mathbf{x}_j,\mathbf{x}_i}$ inherently encompasses the mutual effects between skills. We denote $\alpha_{\{s_1,1\},s_2}$ as the effect on the target skill s_2 caused by answering the question correctly with skill s_1. Similarly, $\alpha_{\{s_1,0\},s_2}$ means the situation when s_1 is incorrect. Intuitively, if s_1 is a prerequisite of s_2: (1) a low knowledge level of s_1 will have negative impacts on s_2; (2) a high mastery of s_2 may indicate having known s_1 well. Correspondingly, $\alpha_{\{s_1,0\},s_2}$ is expected to be small and $\alpha_{\{s_2,1\},s_1}$ should be large. Therefore, for each skill s_i, we define its prerequisite score $r(s_i) \in \mathbb{R}^{|\mathcal{S}|}$ to represent how likely other skills be a prerequisite of s_i:

$$r(s_i) = \text{softmax}\left(\alpha_{\{s_i,1\},s}\right) / \text{softmax}\left(\alpha_{\{s,0\},s_i}\right) , \tag{12}$$

where the softmax aims to normalize the effects among all skills s. Then given any skill s_i, we can get its most probable prerequisites according to prerequisite score $r(s_i)$.

In current education literature, relations among skills are usually manually annotated, which requires a lot of resources and time. The proposed method can serve as a reference and completion to

[2] We do not make it question-specific because this will be too fine-grained to learn meaningful mutual parameters, and skill is comparably a more suitable level.

Table 2: Statistics of the datasets after preprocessing.

Dataset	#student	#question	#skill	#interaction
ASSISTments 09-10	3.7k	16.9k	111	110.2k
ASSISTments 12-13	25.3k	50.9k	245	879.5k
slepemapy.cz	81.7k	2.9k	1473	2877.5k

education experts, which is important in both online education scenario and traditional classroom teaching. The results of skill relations discovery will be presented in Section 5.5.

5 EXPERIMENTS

5.1 Experimental Settings

5.1.1 ***Datasets.*** We use three real-world datasets to validate the effectiveness of our model.

- **ASSISTments 09-10**. [10] ASSISTments is an online tutoring system that teaches and assesses students in mathematics. This dataset is publicly available[3].
- **ASSISTments 12-13**. This dataset comes from the same system as before with different time spans[4].
- **slepemapy.cz**. [26] This dataset is from an online system used for practicing geography and is publicly available[5]. We utilize *place_asked* as the skill identifier. Each skill will have two questions according to the *type*: (1) find the given place on the map; (2) pick the name for the highlighted place.

For each dataset, we discard invalid users with less than 5 interactions and only consider the first 50 interactions for each user because it is more essential to predict performances when there are few user histories. Besides, the timestamp of each interaction is missing in ASSISTments 09-10, hence we assume users answer questions consecutively with a fixed time gap (1 second). After preprocessing, statistics of the three datasets are shown in Table 2.

5.1.2 ***Evaluation Protocols.*** We perform 5-fold cross validation to evaluate all the models, in which folds are split based on users. A validation set is built by extracting 10% of the users from the training set, which is used to tune hyperparameters and perform early stopping. For each sequence, every position except for the first one will be used for training and evaluation. We use *area under the curve* (AUC) as the evaluation metric. The above settings are also adopted in many previous studies [24, 29].

5.1.3 ***Baseline Methods.*** We compare our HawkesKT model to six baseline methods in different aspects. The first three baselines do not incorporate temporal information:

- **IRT** [16]. This is a traditional method based on Item Response Theory, which models characteristics of items and users with two sets of parameters.
- **DKT** [29]. DKT represents a student's knowledge by the hidden states of RNN. Each skill is encoded to a one-hot vector or a low-dimensional embedding.

Table 3: AUC of all methods on the three datasets (higher is better). We conduct 5-fold cross validation and report the average score. The best results are in bold face and the best baseline is underlined. * and ** means our method significantly outperforms the corresponding baseline with $p < 0.05$ and $p < 0.01$, respectively.

Method	*ASSISTments 09-10*	*ASSISTments 12-13*	*slepemapy.cz*
IRT	0.5869**	0.6340**	0.5654**
DKT	0.7515**	0.7308**	0.7423**
SAKT	0.6860**	0.6906**	0.6599**
DKT-Forgetting	0.7540*	0.7462**	0.7498
KTM	0.7425**	0.7535**	0.7407**
AKT-R	0.7474**	0.7555**	0.7454**
HawkesKT	**0.7629**	**0.7676**	**0.7500**

- **SAKT** [25]. This is a recently proposed deep-learning method based on self-attention mechanism.

The rest three baselines consider time-varying effects:

- **DKT-Forgetting** [24]. This is a DKT-based model that considers past trials and time gaps as extra features.
- **KTM** [34]. This method utilizes Factorization Machines to model interactions between features. Here the features we use include question id, skill id, historical responses on different skills, and temporal features in DKT-Forgetting.
- **AKT-R** [11]. This is a attention-based neural network model and attention weights are computed by a distance-aware exponential decay with a global decay rate, which is a state-of-the-art method with temporal information.

We do not include methods based on BKT because they have already been encompassed in the above methods.

5.1.4 ***Parameter Settings.*** We implement HawkesKT and other baselines (except for IRT) in *PyTorch* and the code is publicly available[6]. For a fair comparison, the embedding size and hidden size is fixed to 64 for different models on all the datasets. Early stop is applied if AUC on the validation set does not increase for 5 epochs. The learning rate is tuned between $\{5e^{-3}, 1e^{-3}, 5e^{-4}, 1e^{-4}\}$ and the l2-coefficient is tuned between $\{1e^{-3}, 1e^{-4}, 1e^{-5}, 1e^{-6}, 0\}$. All the model parameters are normally initialized with 0 mean and 0.01 standard deviation.

5.2 Overall Performance

Table 3 shows the performance of all baseline methods and our HawkesKT model. We have the following observations:

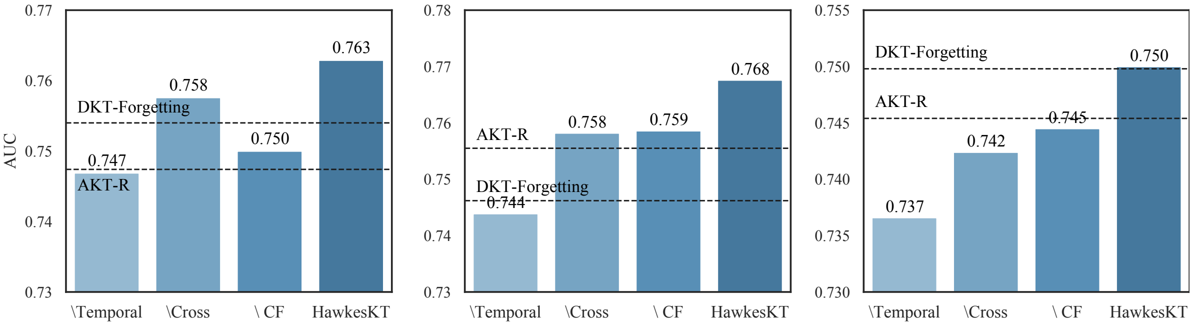
First, different kinds of baselines demonstrate noticeable performance gaps. As a RNN-based model, DKT is superior to the traditional IRT. We also find DKT outperforms SAKT on all datasets, which is consistent with previous work [11]. DKT-Forgetting gain further improvements, showing the importance of considering temporal factors. KTM is flexible to incorporate question-level and

[3] https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010

[4] https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect

[5] https://www.fi.muni.cz/adaptivelearning/?a=data

[6] https://github.com/THUwangcy/HawkesKT



(a) ASSISTments 09-10

(b) ASSISTments 12-13

(c) slepemapy.cz

Figure 4: Performance comparison between HawkesKT and its variants: without temporal information (\Temporal), without adaptive decay rates (\Cross), and without matrix factorization (\CF).

temporal features, and hence perform better than DKT-based models sometimes. AKT-R yields remarkable results most of the time because it not only model temporal decay but also include question-level information by Rasch model-based embeddings. But we find KTM and AKT-R are prone to overfit during training because of the high model capacity.

Second, HawkesKT performs consistently better than all the baselines. Compared with DKT-Forgetting, HawkesKT naturally takes continuous time-varying effects into consideration and does not only rely on interactions with the same skill or adjacent one. KTM is able to extend to incorporate temporal features, but it need hand-crafted features and cannot capture the adaptive temporal cross-effects of each previous interaction. As for AKT-R, although it incorporate exponential decay to model the forgetting behavior, the decay rate is still global. Therefore, it cannot capture temporal cross-effects revealed in this work, and hence results in suboptimal performances. On the contrary, our HawkesKT model addresses temporal cross-effects with mutual excitation and adaptive kernel function, leading to the best results all the time.

Third, our HawkesKT model is capable of scaling to different scenarios. The three datasets involve different subjects, and the sizes of data range from small to large. The consistent improvements demonstrate the scalability of HawkesKT. Note that our model gains more improvements on mathematical datasets. While on the geography dataset (slepemapy.cz), the improvement is not that large (the difference compared to DKT-Forgetting is not significant). It is reasonable because the temporal cross-effects between skills are indeed more helpful in mathematics. As for geography, only the surrounding countries can help identify the target country in general. The relationships between skills are simple and the major temporal dynamic is self-forgetting, which accounts for why DKT-forgetting and AKT-R perform well.

5.3 Efficiency Analyses

As a new branch of method, we also investigate the efficiency issue of HawkesKT. Table 4 shows the training time per epoch and the total number of parameters for different methods on two representative datasets. We ensure all the methods are evaluated under the same experimental setting (batch size, embedding size, maximum sequence length). All the experiments are conducted with a single 1080Ti GPU.

Table 4: Comparison of the training time and number of parameters under the same experimental setting.

Method	*ASSISTments 09-10*		*slepemapy.cz*	
	time/epoch	# params	time/epoch	# params
DKT	0.8s	57.4k	20.4s	314.9k
DKT-Forgetting	1.1s	59.1k	47.6s	320.5k
KTM	7.5s	1760.7k	317.8s	475.4k
AKT-R	2.3s	160.7k	45.8s	649.2k
HawkesKT	**0.5s**	74.8k	**15.2s**	564.6k

We can observe that the training time of HawkesKT is much less than other state-of-the-art methods, even faster than DKT. KTM is especially slow and needs abundant parameters if there are a lot of questions in the dataset. ART-R is also not efficient because of its complex model structure. Remarkably, our HawkesKT not only has fewer parameters compared to recent work, but also reduces the training cost by a large margin while achieving the best performance. In real educational scenarios, timeliness is also an important factor. The significant advantage of HawkesKT in both effectiveness and efficency will make it more applicable in real-world large-scale educational settings.

5.4 Ablation Study

To verify the impacts of modeling temporal cross-effects, we compare HawkesKT with three variants:

- **\Temporal**: This model removes the kernel function, and thus does not consider forgetting behavior, leading to an intensity function as follows: $\lambda(\mathbf{x}_i) = \lambda_0^{\mathbf{x}_i} + \sum_{\mathbf{x}_j \in S_{t_i}} \alpha_{\mathbf{x}_j,\mathbf{x}_i}$.
- **\Cross**: This model uses a global parameter β to control the exponential decay: $\lambda(\mathbf{x}_i) = \lambda_0^{\mathbf{x}_i} + \sum_{\mathbf{x}_j \in S_{t_i}} \alpha_{\mathbf{x}_j,\mathbf{x}_i} e^{-\beta \log(t_i - t_j)}$.
- **\CF**: This model does not use matrix factorization as a reparameterization method, and directly optimizes parameter $\mathbf{A}, \mathbf{B}$ with shape $2|\mathcal{S}| \times |\mathcal{S}|$.

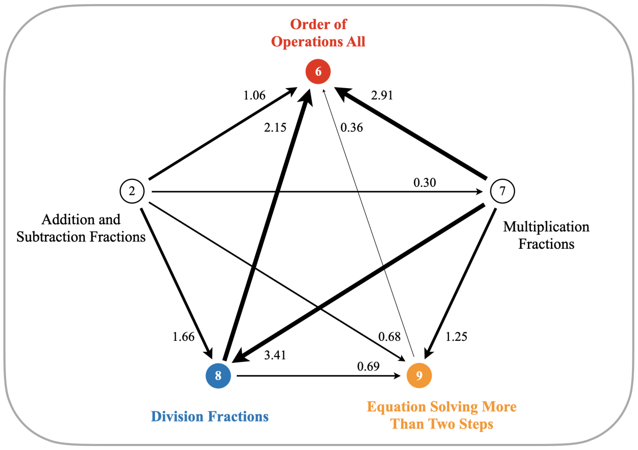


Figure 5: Visualization of the discovered relations between skills according to the learned parameters ($\alpha_{\mathbf{x}_j,\mathbf{x}_i}$). Circles represent skills and arrows are relations. Stronger relations will be showed in thicker arrows, and circles with color are some representative skills worthy of notice.

Figure 4 shows the AUC of HawkesKT and its variants on all the datasets, as well as DKT-Forgetting and AKT-R for comparison. We have the following main observations:

First, temporal information is of significant importance in KT. \Temporal results in the largest performance loss and is generally worse than DKT-Forgetting and AKT-R, which shows the necessity to model forgetting behavior.

Second, it is important to model fine-grained temporal evolution with different decay rates to capture temporal cross-effects in KT. Although the performance loss of \Cross is not the largest, it is noteworthy that \Cross leads to consistently worse results on all the datasets. Without temporal cross-effects, \Cross yields similar results with AKT-R on ASSISTments 12-13. This shows the global decay rate is not sufficient, and the adaptive temporal cross-effects addressed in our model are indeed helpful.

Third, reparameterization with matrix factorization brings stable performance gain. Matrix factorization helps to take advantage of collaborative filtering, which enables the learned patterns to propagate through embeddings and generalize under different circumstances. Without matrix factorization, performances of \CF on all the datasets suffer a moderate loss, which shows the effectiveness of combining point process and collaborative filtering.

5.5 Skill Relationships Discovery

Here we want to validate the performance of relation discovery based on the proposed prerequisite score in Section 4.5.

Firstly, we utilize the parameters trained on ASSISTments 12-13 with top-10 frequency skills as a case study. We visualize the relations among some representative skills in Figure 5. The circles represent skills and arrows stand for prerequisite relations between skills. The calculated prerequisite score is also annotated beside the arrow (the thicker of the arrow, the stronger of the relation). Circles with color are representative skills we focus on, whose main prerequisites, as well as relations among them are drawn. The figure shows that our model indeed finds some meaningful relations. For example, *Order of Operations All* (6 in red) relies on skills about addition/subtraction (2) and multiplication/division (7, 8). The prerequisite scores for multiplication/division skills are higher because they are more essential when determining the order of operations. Besides, *Multiplication Fractions* (7) is a strong prerequisite of *Division Fractions* (8 in blue), and *Addition and Subtraction Fractions* (2) also has some impacts.

Secondly, we conduct an annotation experiment to quantitatively validate the discovered relationships. We choose top-20 frequency skills in ASSISTments 12-13 and ask three experts to annotate the binary helpfulness between each skill pair. The kappa coefficient of annotations is 0.52, showing the applicability. Averaged annotation results are used as the ground truth of relevance. Then a ranked list for each skill is generated based on the proposed prerequisite score. This ranked list is evaluated according to the annotated relevance, whose averaged NDCG is 0.8267. This shows our model can indeed automatically find relationships between skills consistent with human perceptions.

The above analyses validate our parametric assumptions, and demonstrate that the parameters in HawkesKT are highly interpretable. The revealed relation graph can serve as an effective completion for education experts. This method can also scale to find relations among a large amount of skills, which is helpful in both online and traditional education scenarios.

6 CONCLUSION AND FUTURE WORK

In this paper, we propose to explicitly model *temporal cross-effects* in KT, which means the adaptive time-varying effects between different interactions. Through empirical studies, we validate that different cross-skill effects have varying temporal dynamics. Based on the temporal cross-effects shown in data, a novel point process based model HawkesKT is proposed, which reveals a new branch of method for KT. In HawkesKT, each history interaction will have its own continuously changing effect on the target skill, controlled by the corresponding kernel function. The proposed HawkesKT achieves superior performance compared to state-of-the-art methods on three real-world datasets in different scenarios. It is also remarkable that our model shows significant advantages in training efficiency and parameter interpretability. We further propose prerequisite score to automatically discover latent skill relationships based on parameters in our model, which can serve as a reference and completion to experts in education.

In the future, we plan to enable HawkesKT with extensible side information, as the current model is not flexible enough to take other features into consideration, such as school, the type of question, and so on. We also consider incorporating known dependencies among skills to improve the prediction performance.

ACKNOWLEDGMENTS

This work is supported by the National Key Research and Development Program of China (2018YFC0831900), Natural Science Foundation of China (Grant No. 62002191, 61672311, 61532011) and Tsinghua University Guoqiang Research Institute. This project is also funded by China Postdoctoral Science Foundation (2020M670339) and Weizhi Ma is supported by Shuimu Tsinghua Scholar Program.

REFERENCES

[1] Odd Aalen, Ornulf Borgan, and Hakon Gjessing. 2008. *Survival and event history analysis: a process point of view.* Springer Science & Business Media.

[2] Ghodai Abdelrahman and Qing Wang. 2019. Knowledge Tracing with Sequential Key-Value Memory Networks. (2019).

[3] Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International Conference on Intelligent Tutoring Systems*. Springer, 164–175.

[4] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An efficient adaptive transfer neural network for social-aware recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 225–234.

[5] Lap Pong Cheung and Haiqin Yang. 2017. Heterogeneous features integration in deep knowledge tracing. In *International Conference on Neural Information Processing*. Springer, 653–662.

[6] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction* 4, 4 (1994), 253–278.

[7] David Roxbee Cox and Valerie Isham. 1980. *Point processes*. Vol. 12. CRC Press.

[8] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.

[9] Nan Du, Yichen Wang, Niao He, Jimeng Sun, and Le Song. 2015. Time-sensitive recommendation from recurrent user activities. In *Advances in Neural Information Processing Systems*. 3492–3500.

[10] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User Modeling and User-Adapted Interaction* 19, 3 (2009), 243–266.

[11] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-Aware Attentive Knowledge Tracing. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

[12] Robert J Harvey and Allen L Hammer. 1999. Item response theory. *The Counseling Psychologist* 27, 3 (1999), 353–383.

[13] Alan G Hawkes. 1971. Spectra of some self-exciting and mutually exciting point processes. *Biometrika* 58, 1 (1971), 83–90.

[14] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. 2020. Learning or Forgetting? A Dynamic Approach for Tracking the Knowledge Proficiency of Students. *ACM Transactions on Information Systems (TOIS)* 38, 2 (2020), 1–33.

[15] Jussi Kasurinen and Uolevi Nikula. 2009. Estimating programming knowledge with Bayesian knowledge tracing. *ACM SIGCSE Bulletin* 41, 3 (2009), 313–317.

[16] Mohammad M Khajah, Yun Huang, José P González-Brenes, Michael C Mozer, and Peter Brusilovsky. 2014. Integrating knowledge tracing and item response theory: A tale of two frameworks. In *CEUR Workshop Proceedings*, Vol. 1181. University of Pittsburgh, 7–15.

[17] Byung-Hak Kim, Ethan Vizitei, and Varun Ganapathi. 2018. GritNet: Student performance prediction with deep learning. *arXiv preprint arXiv:1804.07405* (2018).

[18] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[19] Yehuda Koren and Robert Bell. 2015. Advances in collaborative filtering. In *Recommender systems handbook*. Springer, 77–118.

[20] Amar Lalwani and Sweety Agrawal. 2019. What Does Time Tell? Tracing the Forgetting Curve Using Deep Knowledge Tracing. In *International Conference on Artificial Intelligence in Education*. Springer, 158–162.

[21] David Marsan and Olivier Lengline. 2008. Extending earthquakes' reach through cascading. *Science* 319, 5866 (2008), 1076–1079.

[22] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*. 6754–6764.

[23] Hongyuan Mei and Jason M Eisner. 2017. The neural hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems*. 6754–6764.

[24] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In *The World Wide Web Conference*. ACM, 3101–3107.

[25] Shalini Pandey and George Karypis. 2019. A Self-Attentive model for Knowledge Tracing. *arXiv preprint arXiv:1907.06837* (2019).

[26] Jan Papoušek, Radek Pelánek, and Vít Stanislav. 2016. Adaptive geography practice data set. *Journal of Learning Analytics* 3, 2 (2016), 317–321.

[27] Philip I Pavlik Jr, Hao Cen, and Kenneth R Koedinger. 2009. Performance Factors Analysis–A New Alternative to Knowledge Tracing. *Online Submission* (2009).

[28] Radek Pelánek. 2015. Modeling Students' Memory for Application in Adaptive Educational Systems. *International Educational Data Mining Society* (2015).

[29] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *Advances in neural information processing systems*. 505–513.

[30] Yumeng Qiu, Yingmei Qi, Hanyuan Lu, Zachary A Pardos, and Neil T Heffernan. 2011. Does Time Matter? Modeling the Effect of Time with Bayesian Knowledge Tracing.. In *EDM*. 139–148.

[31] Martin Schittek, Nikos Mattheos, HC Lyon, and Rolf Attström. 2001. Computer assisted learning. A review. *European Journal of Dental Education: Review Article* 5, 3 (2001), 93–100.

[32] Yu Su, Qingwen Liu, Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Chris Ding, Si Wei, and Guoping Hu. 2018. Exercise-enhanced sequential modeling for student performance prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[33] Yusuke Tanaka, Takeshi Kurashima, Yasuhiro Fujiwara, Tomoharu Iwata, and Hiroshi Sawada. 2016. Inferring latent triggers of purchases with consideration of social effects and media advertisements. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*. ACM, 543–552.

[34] Jill-Jênn Vie and Hisashi Kashima. 2019. Knowledge Tracing Machines: Factorization Machines for Knowledge Tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 750–757.

[35] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2019. Modeling Item-Specific Temporal Dynamics of Repeat Consumption for Recommender Systems. In *The World Wide Web Conference*. ACM, 1977–1987.

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

[37] Yutao Wang and Neil T Heffernan. 2012. Leveraging First Response Time into the Knowledge Tracing Model. *International Educational Data Mining Society* (2012).

[38] Shuai Xiao, Junchi Yan, Changsheng Li, Bo Jin, Xiangfeng Wang, Xiaokang Yang, Stephen M Chu, and Hongyuan Zha. 2016. On Modeling and Predicting Individual Paper Citation Count over Time.. In *IJCAI*. 2676–2682.

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

[40] Ke Zhou, Hongyuan Zha, and Le Song. 2013. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*. 1301–1309.