Learning Behavior-oriented Knowledge Tracing

Bihan Xu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
xbh0720@mail.ustc.edu.cn

Zhenya Huang*
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
huangzhy@ustc.edu.cn

Jiayu Liu
School of Data Science, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
jy251198@mail.ustc.edu.cn

Shuanghong Shen
School of Data Science, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
closer@mail.ustc.edu.cn

Qi Liu
School of Computer Science and Technology, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
qiliuql@ustc.edu.cn

Enhong Chen
Anhui Province Key Laboratory of Big Data Analysis and Application, University of Science and Technology of China & State Key Laboratory of Cognitive Intelligence
Hefei, China
cheneh@ustc.edu.cn

Jinze Wu
iFLYTEK Research, iFLYTEK Co., Ltd
Hefei, China
hxwjz@mail.ustc.edu.cn

Shijin Wang
State Key Laboratory of Cognitive Intelligence & iFLYTEK AI Research, iFLYTEK Co., Ltd
Hefei, China
sjwang3@iflytek.com

ABSTRACT

Exploring how learners' knowledge states evolve during the learning activities is a critical task in online learning systems, which can facilitate personalized services downstream, such as course recommendation. Most of existing methods have devoted great efforts to analyzing learners' knowledge states according to their responses (i.e., right or wrong) to different questions. However, the significant effect of learners' learning behaviors (e.g., answering speed, the number of attempts) is omitted, which can reflect their knowledge acquisition deeper and ensure the reliability of the response. In this paper, we propose a *L*earning *B*ehavior-oriented *K*nowledge *T*racing (LBKT) model, with the goal of explicitly exploring the learning behavior effects on learners' knowledge states. Specifically, we first analyze and summarize several dominated learning behaviors including *Speed*, *Attempts* and *Hints* in the learning process. As the characteristics of different learning behaviors vary greatly, we separately estimate their various effects on learners' knowledge acquisition in a quantitative manner. Then, considering that different learning behaviors are closely dependent with each other, we assess the fused effect of multiple learning behaviors by capturing their complex dependent patterns. Finally, we integrate the forgetting factor with learners' knowledge acquisition to comprehensively update their changing knowledge states in learning. Extensive experimental results on several public datasets demonstrate that our model generates better performance prediction for learners against existing methods. Moreover, LBKT shows good interpretability in tracking learners' knowledge state by incorporating the learning behavior effects. Our codes are available at https://github.com/xbh0720/LBKT.

CCS CONCEPTS

• **Information systems** → **Data mining**; • **Social and professional topics** → **Student assessment**; • **Applied computing** → **E-learning**.

KEYWORDS

Knowledge Tracing, Learning Behaviors, User Modeling

ACM Reference Format:
Bihan Xu, Zhenya Huang, Jiayu Liu, Shuanghong Shen, Qi Liu, Enhong Chen, Jinze Wu, and Shijin Wang. 2023. Learning Behavior-oriented Knowledge Tracing. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3580305.3599407

*Zhenya Huang is the corresponding author.

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599407

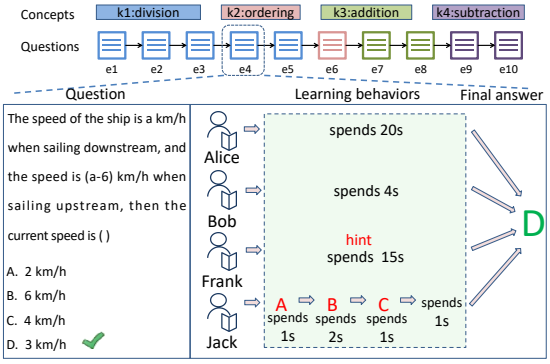


Figure 1: A toy example of four learners' learning behaviors when answering the same question e_4 on the online learning system. Although all the learners finally choose the correct option "D", they show extremely different behaviors.

1 INTRODUCTION

Online learning platforms, such as MOOC, KhanAcedemy.org, have shown an increasing attention for learners nowadays due to their convenience of accessing massive learning resources [3, 12, 26, 43, 47]. One of the dominated research topics in online learning is knowledge tracing (KT), which aims to estimate and track learners' knowledge states on different concepts (e.g., Function) from their academic performance on answering questions during their learning activities [28]. On the basis of the KT research, learners can realize their weakness and save energy to prepare the targeted practices. Besides, systems can tailor several personalized services, such as arranging appropriate learning path and recommending suitable learning items, which help avoid duplicated trials [11, 13, 48, 56].

In the literature, many works have been devoted to KT tasks ranging from the earlier representative Bayesian knowledge tracing based models to recent deep knowledge tracing based advances [17, 37, 52]. Generally, existing models try every effort to assess the learning results of question-answering sequences, where learners' responses (i.e., right or wrong) as well as the corresponding question information (e.g., knowledge concept, question text) have been explored as much as possible [27, 35, 54]. However, they are insufficient in exploitation of the learners' critical learning behaviors [55]. As shown in Figure 1, four learners are required to solve the same question e_4 at step 4, and all of them choose the correct option "D" as final answers. Focusing on this example, if we only assess whether they could answer correctly during learning, we can conclude that these four learners have equally benefited from this question-solving process. However, this conclusion would be untenable if we further analyze their learning behaviors when answering. For example, Alice spends 20s to choose "D" but Bob only consumes 4s. Taking this evidence into consideration, we tend to conclude that Alice and Bob should have quite different fine-grained knowledge acquisition on the related "division" concept. Besides, Frank requested a hint, while Jack attempted four times, their responses should have different reliabilities compared with Alice and Bob. Therefore, we argue that monitoring the detailed learning behaviors of learners is essential for the KT research.

In recent years, some researchers have noticed the importance of the behavior information and utilized it to enhance the KT task [40, 42, 51, 55]. However, they generally applied learners' behaviors as additional features with the model input explicitly or implicitly. Actually, learning behaviors have quite complex effects during learners' learning process, which remain unexplored. We investigate and summarize three typical learning behaviors which dominate how learners perform on questions and how much knowledge they could acquire: **(1) Speed**. The speed of answering has been confirmed to be closely related to learners' knowledge states [24, 49]. For example, a short initial response time could indicate either high proficiency or "gaming" behavior (e.g., Bob may be well mastered or just guess the answer in Figure 1) [4]. In contrast, a long initial response time could be caused by either careful thinking or lack of concentration. **(2) Attempts**. The number of learners' attempts on a specific question is another important factor. Specifically, a learner could learn quickly from multiple attempts on the same question related to unknown concepts. However, it is also possible that the learner just impatiently treats the question and attempts repeatedly for correct answers which leads to poor knowledge gain subsequently [38]. **(3) Hints**. Online learning systems will offer hints for learners who are lack of knowledge mastering when practicing [2]. Learners can obtain inspiration from these hints and achieve promotion but some may choose to directly access the correct answer through abusing hints instead of making efforts on their own and thus resulting in poor knowledge acquisition [4, 10].

In this paper, we aim to comprehensively assess the complex but significant effects of the above three learning behaviors on tracking learners' knowledge states, where we try to figure out the following challenges. First, as we have mentioned above, previous pedagogy studies indicated that learning behaviors have complex effect mechanisms on the knowledge acquisition [6, 7, 24, 38]. Moreover, different behavioral characteristics vary greatly, so it is difficult to quantify the distinctive effects of each behavior on assessing learners' knowledge acquisition during the learning process. Second, learning behaviors do not individually produce effects, but are closely dependent with each other. For example, looking at the behavior of *Attempts* alone, we are not sure whether Jack who makes multiple attempts in Figure 1 is learning through repeated trials or impatiently attempting to try out the correct answer. But when combining his rapid speed during each attempt, we are more confident that the latter is the case. Therefore, by considering learners' different behaviors together, we can make more reliable assessment of their knowledge states. However, it is a great challenge to capture the complex dependent patterns of multiple behaviors. Third, besides knowledge acquisition, learners' tendency of forgetting also cannot be ignored, considering behavior effects on both knowledge acquisition and forgetting would bring many obstacles on how to update the knowledge state of learners during the process.

To address these challenges, we propose a novel *L*earning *B*ehavior-oriented *K*nowledge *T*racing (LBKT) model, which tries to explore how learning behaviors affect learners' knowledge states in an explicit way. Specifically, we first analyze the distribution of the learning behaviors including *Speed*, *Attempts*, and *Hints*, and accordingly assess their separate effects on knowledge acquisition with a quantitive estimation. Then, we design the Fused Behavior Effect Measuring module to capture the dependency among different behaviors by modeling their high-order interaction patterns. Subsequently, we update the knowledge states of learners with a

novel delicate forget gate considering both the decline of memory and the stimulation of knowledge acquisition on the evolution process. Finally, the experiments results on several public datasets demonstrate that our LBKT not only generates better performance predictions for learners but also shows the superior interpretability in presenting learners' knowledge proficiency considering the learning behavior effects. To the best of our knowledge, this is first few attempt to go deeper for exploring knowledge tracing with quantitively modeling the complex learning behavior effects.

2 RELATED WORK

Knowledge Tracing. Knowledge Tracing (KT) is an essential task which aims at tracking learners' knowledge state dynamically in the online learning system. Existing approaches can be categorized as traditional methods and deep learning methods [31]. Bayesian Knowledge Tracing (BKT) is one of the most popular traditional methods, which assessed the learner's proficiency on different concepts in a Hidden Markov Model. Thai-Nghe et al. [44] used the Tensor Factorization method to project learners into a latent space on different time steps. Huang et al. [22] further considered the learning theory and Ebbinghaus forgetting curve in constraining the transition of learners' knowledge matrix. With the development of deep learning, DKT first introduced recurrent neural network to model learners' knowledge state [37]. Different from DKT using one hidden high-dimensional vector to represent the knowledge state, DKVMN facilitated a dynamic value memory matrix to store and update learners' proficiency on each concept [54]. Shen et al. [41] applied a convolutional sliding window to model learners' individualized learning. SAKT first introduced self-attention mechanism into KT to capture the dependency among the learning sequence [36]. AKT further used the Rasch model to combine questions' individualized difficulty with concepts for better question representation [17]. Liu et al. [27] utilized text information to enrich the question's embedding. Furthermore, some work model the constraint or propagation of knowledge proficiency among related concepts with their structure information [9, 35].

Some research has noticed the importance of learning behaviors and applied them to KT models. Schultz and Arroyo [39] believed that learners' performance was determined by both their evolving knowledge states and motivation reflected in the learning behaviors. Wang and Heffernan [49] thought that first response time could help predict learners' performance and combined the prediction result with existing KT models. Besides, some studies utilized learners' engagement in multiple learning materials like video lectures and question-related hints to better evaluate their knowledge proficiency [1, 46, 57]. For example, Mongkhonvanit et al. [32] introduced learners' behaviors like playback speed and fast-forwarded when they watched videos in the MOOC into DKT's input. LPKT made a finer application of behaviors that considered the influence of interval and answer time on learners' learning and forgetting in the learning process [40]. However, these models just concatenate the behavioral features with the input of a neural network as supplementary information without fully mining the way how learning behaviors affect learners' knowledge state.

Learning Behaviors Analysis. There are many pieces of research about learning behavior analysis [33, 55], which mainly focus on two aspects. One is using learning behaviors to assess learners' motivation or attitude during learning. Vicente and Pain [45] classified learners' motivational state into 9 axes such as "Fantasy" and "Satisfaction" according to their interactions when learning Japanese numbers. Joseph [24] used response time to catch learner's disengagement and presented the positive correlation between response time and performance. Baker et al. [5] utilized multiple behaviors to detect gaming and made timely intervention to prevent learners from gaming. The other aspect is using learning behaviors to better assess learners' proficiency [16, 19]. Some cognitive diagnosis models predict learners' performance based on their motivations. For example, Wu et al. [50] used learners' multiple-attempt response to capture their gaming probability according to four assumptions (e.g. the less time, the higher gaming factor) and based on which predicted learners' performance thus obtaining more accurate knowledge proficiency estimation. Similarly, Johns and Woolf [23] applied HMM to dynamically track learners' motivation as "unmotivated-hint", "unmotivated-guess" and "motivated" according to the requested-hints and response time behaviors.

In summary, learning behaviors are confirmed to be significant in analyzing learners' learning process and show complex effects on their knowledge states. In contrast to previous works that insufficiently view behaviors as additional features, our work deeply mine how the multiple learning behaviors produce effect.

Table 1: Notations and descriptions.

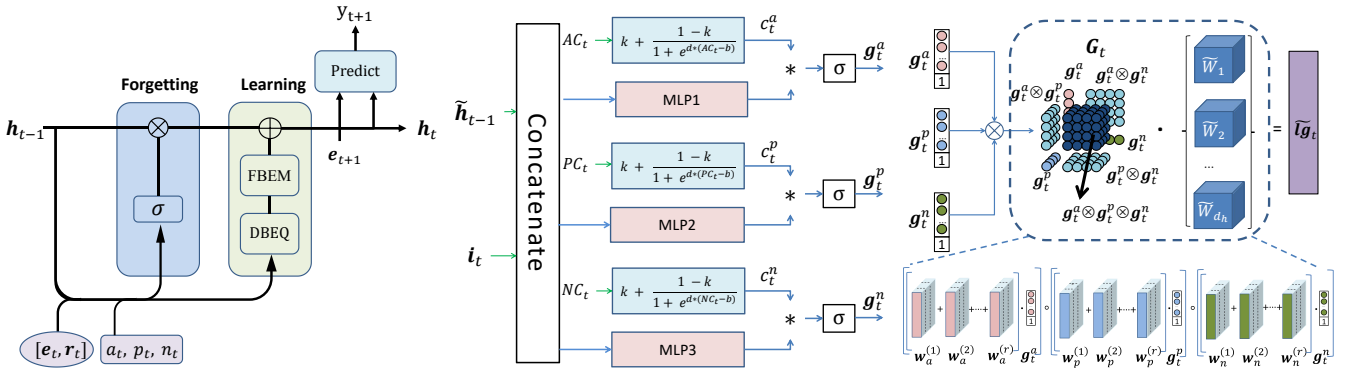
Notations	Descriptions
X	Learners' learning sequence.
$\boldsymbol{h}_t$	Learners' knowledge state matrix.
c_m	The knowledge concept.
e_t, $\boldsymbol{e}_t$	The question and its embedding.
r_t, $\boldsymbol{r}_t$	The response and its embedding.
$\boldsymbol{i}_t$	The embedding of a basic question-answer interaction.
Q, $\boldsymbol{q}_{e_t}$	The question-concept relation matrix and one row in it.
y_t	The prediction of performance.
a_t, p_t, n_t	The behaviors of *Speed*, *Attempts* and *Hints*.
AC_t, PC_t, NC_t	The rescaled factors generated from learning behaviors.

3 PROBLEM DEFINITION

In this section, we give a formal definition of our task and summarize the notations used throughout the paper in Table 1.

Supposing there are $|S|$ learners and $|E|$ questions in the online learning system. Generally, the learning result sequence of a specific learner is recorded as $X = \{(e_1, r_1), (e_2, r_2), ..., (e_T, r_T)\}$, where $e_t \in E$ denotes the question answered by the learner at time step t, and r_t is the corresponding response (1 means correct and 0 for incorrect). Since behaviors affect learners' knowledge acquisition, using only the basic step of learning results (e_t, r_t) is insufficient to capture the evolution of knowledge state. In order to better assess learners' knowledge state, we consider three dominated behaviors of *Speed*, *Attempts* and *Hints* in the paper. Thus, the learning sequence including the behaviors is represented as $X = \{(e_1, r_1, b_1), (e_2, r_2, b_2), ..., (e_T, r_T, b_T)\}$, where b_t denotes the learning behaviors, composed of (a_t, p_t, n_t) which represent *Speed*, *Attempts*, and *Hints* respectively. After taking the learning behaviors into account, the definition of our task is:

Given learners' learning records $X = \{(e_1, r_1, b_1), (e_2, r_2, b_2), ..., (e_T, r_T, b_T)\}$*, we aim to dynamically track learners' knowledge state and predict their performance on future questions.*



(a) The framework of LBKT. (b) Differentiated Behavior Effect Quantifying (DBEQ). (c) Fused Behavior Effect Measuring (FBEM).

Figure 2: The framework and details of LBKT.

4 THE LBKT MODEL

Figure 2(a) illustrates the overview of LBKT framework. We propose the Differentiated Behavior Effect Quantifying (DBEQ) module and Fused Behavior Effect Measuring (FBEM) module to evaluate the distinctive and cooperative effect of learning behaviors on learners' knowledge acquisition. Then a forget gate incorporated with learning behaviors is proposed to model the decline of proficiency over time. By combining the forgetting factor and knowledge acquisition together, the knowledge state is updated.

4.1 Learning Sequence Representation

Knowledge State Embedding. Following existing work [27, 40], we represent the knowledge state by a matrix $\boldsymbol{h} \in \mathbb{R}^{M \times d_h}$, where M denotes the number of concepts and d_h is the dimension. Therefore, each row in $\boldsymbol{h}$ represents the proficiency of the corresponding concept and will be updated in the training process. $\boldsymbol{h}_t$ is utilized to represent the knowledge state at step t.

Basic Interaction Embedding. We treat the question-answer pair in a step as a basic interaction. A question embedding matrix $\boldsymbol{E}_1 \in \mathbb{R}^{|E| \times d_e}$ is utilized to embed the question set where $|E|$ denotes the number of questions and d_e is the dimension. Then, the question answered at time step t is represented as $\boldsymbol{e}_t \in \mathbb{R}^{d_e}$ by looking up $\boldsymbol{E}_1$ with the question id. For the answer, we employ another embedding matrix $\boldsymbol{E}_2 \in \mathbb{R}^{2 \times d_a}$ to encode the two kinds of response results (correct or incorrect) into a d_a-dimensional vector. And the response at time step t is denoted as $\boldsymbol{r}_t$. To obtain the basic interaction embedding, we use a fully connected layer to deeply fuse the question and answer embedding as follows:

$$\boldsymbol{i}_t = ReLU(\boldsymbol{W}_1[\boldsymbol{e}_t \oplus \boldsymbol{r}_t] + \boldsymbol{b}_1), \tag{1}$$

where $\boldsymbol{W}_1 \in \mathbb{R}^{d_h \times (d_e + d_a)}$, $\boldsymbol{b}_1 \in \mathbb{R}^{d_h}$ are the trainable parameters, $\oplus$ means the concatenate operation.

Q-matrix. We use a Q-matrix $\boldsymbol{Q} \in \mathbb{R}^{|E| \times M}$ in which each element is 0 or 1 to represent the relationship between questions and concepts [8]. $\boldsymbol{Q}_{jm} = 1$ means that the question e^j is related to concept c_m and 0 otherwise. $\boldsymbol{q}_{e_t}$ is a row in $\boldsymbol{Q}$ which represents the related concepts vector of question e_t. According to pedagogical theory [15], the performance on a specific question will only influence the knowledge proficiency on its corresponding concepts. Therefore, when a learner finishes answering question e_t at time t, the update of his knowledge state matrix $\boldsymbol{h}_{t-1}$ is controlled by $\boldsymbol{q}_{e_t}$. We represent the knowledge state on the related concepts of e_t as:

$$\widetilde{\boldsymbol{h}}_{t-1} = \boldsymbol{q}_{e_t} \cdot \boldsymbol{h}_{t-1}. \tag{2}$$

4.2 Differentiated Behavior Effect Quantifying

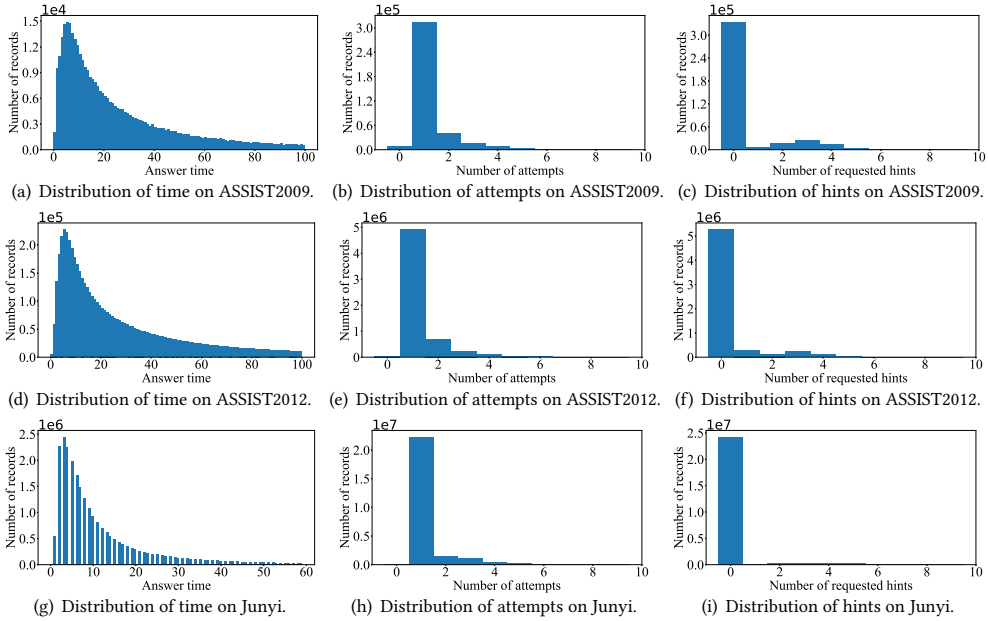
Generally, different learning behaviors have quite different characteristics and affect learners' knowledge acquisition in different ways. In this section, we first measure the differentiated effects of three specific learning behaviors: *Speed*, *Attempts*, and *Hints*. Noting that there are many other learning behaviors you can think of, such as *Modifying* and *Discussing*. As this paper mainly focuses on exploring the learning behavior effects on learners' knowledge states, we choose the above three dominant learning behaviors according to the suggestions of existing works [28, 39, 50] and leave the extension to more kinds of learning behaviors as future work.

Specifically, each learning behavior has complex effects on learners' knowledge acquisition. For example, a high speed may represent proficiency or guessing and a medium speed correlates to carefully thinking in most cases. To tackle this problem, we first analyze the data distribution of these behaviors on three widely-used real-world datasets: ASSIST2009, ASSIST2012, and Junyi. The details of the datasets are given in Section 5.1. Figure 3 presents the distributions of different behaviors on three datasets, based on which, we then can quantify their differentiated effects accordingly.

Speed Effect. As discussed in Section 1, the speed of answering closely reflects learners' knowledge levels. We use the feature of "first response time" in the datasets which denotes the answer time of learner's first attempt to represent *Speed*. Figure 3(a), 3(d), and 3(g) present the distribution of learners' answer time, where the number of records approximately shows the form of log-normal distribution with respect to the answer time (in seconds). Therefore, we assume that the answer time a_{ji} of learner i on question e^j obeys $\ln a_{ji} \sim \mathcal{N}(\mu_j, \sigma_j^2)$, where μ_j, σ_j can be learned by the maximum-likelihood estimation (MLE) method from the records. On this basis, we compute the *Speed* factor AC_{ji} as:

$$AC_{ji} = 1 - P(\mathcal{N}(\mu_j, \sigma_j^2) \leq \ln a_{ji}), \tag{3}$$

where higher speed correlates to higher AC_{ji}. Then, we quantify the distinctive impact of *Speed* factor and obtain the knowledge acquisition vector $\boldsymbol{g}_t^a$ monitored by *Speed* factor AC_t at time step t



(a) Distribution of time on ASSIST2009. (b) Distribution of attempts on ASSIST2009. (c) Distribution of hints on ASSIST2009.

(d) Distribution of time on ASSIST2012. (e) Distribution of attempts on ASSIST2012. (f) Distribution of hints on ASSIST2012.

(g) Distribution of time on Junyi. (h) Distribution of attempts on Junyi. (i) Distribution of hints on Junyi.

Figure 3: The distribution of learning behaviors data on datasets ASSIST2009, ASSIST2012 and Junyi.

as follows:

$$c_t^a = k + \frac{1-k}{1+e^{d\cdot(AC_t-b)}},$$
$$\boldsymbol{g}_t^a = \sigma(c_t^a \cdot (\boldsymbol{W}_2^a[\widetilde{\boldsymbol{h}}_{t-1} \oplus \boldsymbol{i}_t] + \boldsymbol{b}_2^a)), \tag{4}$$

where $\sigma(\cdot)$ is the sigmoid function, $\boldsymbol{W}_2^a \in \mathbb{R}^{d_h \times 2d_h}$, $\boldsymbol{b}_2^a \in \mathbb{R}^{d_h}$ are trainable parameters. c_t^a is the controlling factor extracted from AC_t by a non-linear curve and utilized to control how much knowledge learners could acquire. We choose the non-linear curve inspired by IRT [14] that predicts learners' performance by distinguishing their latent proficiency and similarly we employ it to distinguish the *Speed* factor. k, d, and b are hyper-parameters determining the curve's shape, and we will analyze their influence in Section 5.6.

Attempts Effect. To analyze the attempt effect, we examine the data distribution of "attempt count" attribute in the datasets, representing the number of attempts learners used, to summarize the characteristics of *Attempts*. As shown in Figure 3(b), 3(e), and 3(h), most learners attempted only once, consistent with the common sense. However, there are still some learners who require repeated attempts, either to learn unknown concepts from or to impatiently try out the correct answer. And their knowledge growth should be significantly different from those who attempted only once [7, 38]. As the number of attempts obeys the Poisson distribution approximately, we assume $p_{ji} \sim \mathcal{P}(\lambda_j^p)$ where p_{ji} denotes the number of attempts learner i uses on question e^j. The parameter λ_j^a can be learned by MLE. Then we compute the *Attempts* factor as:

$$PC_{ji} = 1 - P(\mathcal{P}(\lambda_j^p) \geq p_{ji}), \tag{5}$$

where more attempts stand for the higher PC_{ji}. Similar to *Speed*, we compute the knowledge acquisition vector affected by PC_t as:

$$c_t^p = k + \frac{1-k}{1+e^{d\cdot(PC_t-b)}},$$
$$\boldsymbol{g}_t^p = \sigma(c_t^p \cdot (\boldsymbol{W}_2^p[\widetilde{\boldsymbol{h}}_{t-1} \oplus \boldsymbol{i}_t] + \boldsymbol{b}_2^p)), \tag{6}$$

Hints Effect. In terms of the hints effect, learners generally answer questions without requesting for any hint in most cases as shown in Figure 3(c), 3(f), and 3(i), i.e., the distribution of learners' requested hints number. However, there are also some learners need to learn from hints and achieve promotion, while a few even abuse hints to directly access the correct answer thus leading to poor knowledge acquisition [2]. To capture such differentiation of requested hints number, we assume that $n_{ji} \sim \mathcal{P}(\lambda_j^n)$, where n_{ji} denotes the number of hints learner i uses on question e^j and λ_j^n is also a learnable parameter. Then we define the *Hints* factor as:

$$NC_{ji} = 1 - P(\mathcal{P}(\lambda_j^n) \geq n_{ji}), \tag{7}$$

where more hints used equal to higher NC_{ji}. The knowledge acquisition vector $\boldsymbol{g}_t^n$ affected by NC_t is computed similarly as above:

$$c_t^n = k + \frac{1-k}{1+e^{d\cdot(NC_t-b)}},$$
$$\boldsymbol{g}_t^n = \sigma(c_t^n \cdot (\boldsymbol{W}_2^n[\widetilde{\boldsymbol{h}}_{t-1} \oplus \boldsymbol{i}_t] + \boldsymbol{b}_2^n)), \tag{8}$$

Thus, the knowledge acquisition vectors considering the distinctive effect of each behavior are obtained.

4.3 Fused Behavior Effect Measuring

In practice, different behaviors are not independent with each other but cooperative, and we can make more accurate assessment from the perspective of all of them. For example, if a learner answers a question with a high speed, it does not necessarily mean that the learner is disengaged and maybe he is just so skilled in the corresponding concept. However, if he accompanies with other behaviors like repeated attempts, the confidence for him to guess at a random by quick attempts is higher. Therefore, it's of great importance to capture the complex dependency patterns among different behaviors and model their cooperative effect on learners' knowledge acquisition. Inspired by the Tensor Fusion Network [53], we first concatenate the knowledge acquisition vectors $\boldsymbol{g}_t^a$,

$\boldsymbol{g}_t^p$, $\boldsymbol{g}_t^n$ which are estimated under each dominant behavior from DBEQ with the number 1 as shown in Figure 2(c). Then we make an outer-product operation for these three vectors to get a matrix that keeps the information of each feature as well as brings in the interactive information of every two or three features:

$$\boldsymbol{G}_t = \begin{bmatrix} \boldsymbol{g}_t^a \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{g}_t^p \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{g}_t^n \\ 1 \end{bmatrix}, \tag{9}$$

where $\boldsymbol{G}_t$ is composed of the original features $\boldsymbol{g}_t^a$, $\boldsymbol{g}_t^p$, $\boldsymbol{g}_t^n$, the interactions between two features $\boldsymbol{g}_t^a \otimes \boldsymbol{g}_t^p$, $\boldsymbol{g}_t^a \otimes \boldsymbol{g}_t^n$, $\boldsymbol{g}_t^p \otimes \boldsymbol{g}_t^n$, and the interactions among three features $\boldsymbol{g}_t^a \otimes \boldsymbol{g}_t^p \otimes \boldsymbol{g}_t^n$ as shown in Figure 2(c). As a result, it can well capture the complex interaction of knowledge acquisition vectors estimated from three behaviors. Then, we utilize a fully connected layer to obtain the final knowledge acquisition vector considering the cooperative effect:

$$\widetilde{\boldsymbol{lg}}_t = ReLU(\boldsymbol{W}_3 \cdot \boldsymbol{G}_t + \boldsymbol{b}_3), \tag{10}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d_h \times (d_h+1) \times (d_h+1) \times (d_h+1)}$, $\boldsymbol{b}_3 \in \mathbb{R}^{d_h}$ are trainable parameters. However, the space complexity of $\boldsymbol{G}_t$ and the weight matrix $\boldsymbol{W}_3$ and computation complexity of their multiplication are very high, which increase exponentially with the number of input features. In order to save the memory and computation overhead, we follow LMF [30] to decompose $\boldsymbol{W}_3$. Assuming that $\boldsymbol{W}_3$ is staked by $\widetilde{\boldsymbol{W}}_k \in \mathbb{R}^{(d_h+1) \times (d_h+1) \times (d_h+1)}$, $k = 1, ..., d_h$, each of which will take part in computing one dimension in $\widetilde{\boldsymbol{lg}}_t$. For an order-3 tensor $\widetilde{\boldsymbol{W}}_k$, we can decompose it into vectors in the form of:

$$\widetilde{\boldsymbol{W}}_k = \sum_{i=1}^{R} (\boldsymbol{w}_{a,k}^{(i)} \otimes \boldsymbol{w}_{p,k}^{(i)} \otimes \boldsymbol{w}_{n,k}^{(i)}), \tag{11}$$

where R is the smallest number to make the decomposition valid, and $\boldsymbol{w}_{a,k}^{(i)}, \boldsymbol{w}_{p,k}^{(i)}, \boldsymbol{w}_{n,k}^{(i)} \in \mathbb{R}^{d_h+1}$. We use a fixed rank r to implement R in this paper. Through Eq. (11), the space complexity of $\boldsymbol{W}_3$ reduces from $O(d_h \times (d_h+1) \times (d_h+1) \times (d_h+1))$ to $O(3r \times d_h \times (d_h+1))$.

Let $\boldsymbol{w}_a^{(i)} = [\boldsymbol{w}_{a,1}^{(i)}, \boldsymbol{w}_{a,2}^{(i)}, \cdots, \boldsymbol{w}_{a,d_h}^{(i)}]$, the same applies to $\boldsymbol{w}_p^{(i)}$ and $\boldsymbol{w}_n^{(i)}$, then the high-order matrix multiplication could be transferred:

$$\begin{aligned} \boldsymbol{W}_3 \cdot \boldsymbol{G}_t &= (\sum_{i=1}^{r} (\boldsymbol{w}_a^{(i)} \otimes \boldsymbol{w}_p^{(i)} \otimes \boldsymbol{w}_n^{(i)})) \cdot \boldsymbol{G}_t \\ &= \sum_{i=1}^{r} (\boldsymbol{w}_a^{(i)} \otimes \boldsymbol{w}_p^{(i)} \otimes \boldsymbol{w}_n^{(i)}) \cdot \boldsymbol{G}_t \\ &= \sum_{i=1}^{r} (\boldsymbol{w}_a^{(i)} \otimes \boldsymbol{w}_p^{(i)} \otimes \boldsymbol{w}_n^{(i)}) \cdot (\begin{bmatrix} \boldsymbol{g}_t^a \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{g}_t^p \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \boldsymbol{g}_t^n \\ 1 \end{bmatrix}) \\ &= (\sum_{i=1}^{r} \boldsymbol{w}_a^{(i)} \cdot \begin{bmatrix} \boldsymbol{g}_t^a \\ 1 \end{bmatrix}) * (\sum_{i=1}^{r} \boldsymbol{w}_p^{(i)} \cdot \begin{bmatrix} \boldsymbol{g}_t^p \\ 1 \end{bmatrix}) * (\sum_{i=1}^{r} \boldsymbol{w}_n^{(i)} \cdot \begin{bmatrix} \boldsymbol{g}_t^n \\ 1 \end{bmatrix}), \end{aligned} \tag{12}$$

where $*$ denotes the element-wise multiplication. We transfer the high-order matrix multiplication into low-rank vectors' multiplication with each feature vector thus reducing the computation complexity from $O(d_h \times (d_h+1) \times (d_h+1) \times (d_h+1))$ to $O(r \times d_h \times (d_h+1))$.

As $\widetilde{\boldsymbol{lg}}_t$ is correlated to specific concepts, we need to broadcast it to all concepts. We multiply the concept-focused acquisition vector $\widetilde{\boldsymbol{lg}}_t$ with the concept-related vector $\boldsymbol{q}_{e_t}$:

$$\boldsymbol{lg}_t = \boldsymbol{q}_{e_t}^T \cdot \widetilde{\boldsymbol{lg}}_t. \tag{13}$$

4.4 Knowledge State Updating

In the learning process, learners acquire knowledge through answering questions and their knowledge states change accordingly. Following the intuition, we have modeled the knowledge acquisition of learners monitored by the learning behaviors' effect. However, learners' proficiency will decrease over time as well, and the update of their knowledge states is under the influence of both the forgetting factor and knowledge acquisition [34]. Thus, a forget gate is proposed to model the forgetting phenomenon, in which we hold that a learner's forgetting is closely related to his learning behaviors. For instance, a learner exhibiting quickly repeated attempts will forget more as he is not indeed recalling and applying his knowledge. Therefore, we combine learners' learning behaviors, previous knowledge state and current basic interaction into the forget gate to simulate their forgetting process. By combining the forgetting gate and knowledge acquisition vector, we finish the update of knowledge state as follows:

$$\begin{aligned} \boldsymbol{f_t} &= \sigma(\boldsymbol{W}_4[\boldsymbol{h}_{t-1} \oplus \boldsymbol{i}_t \oplus \boldsymbol{AC}_t \oplus \boldsymbol{PC}_t \oplus \boldsymbol{NC_t}] + \boldsymbol{b}_4), \\ \boldsymbol{h}_t &= \boldsymbol{f}_t * \boldsymbol{h}_{t-1} + \boldsymbol{lg}_t, \end{aligned} \tag{14}$$

where $\boldsymbol{W}_4 \in \mathbb{R}^{d_h \times (2d_h+3d_a)}$, $\boldsymbol{b}_4 \in \mathbb{R}^{d_h}$ are trainable parameters.

4.5 Performance Prediction

Considering that in real learning scenarios, when given a question e_{t+1} at time step $t+1$, learners will use their knowledge of the corresponding concepts to solve the problem. So we first use the concept-related vector $\boldsymbol{q}_{e_{t+1}}$ to extract the corresponding knowledge proficiency $\widetilde{\boldsymbol{h}}_t$ from the knowledge state matrix $\boldsymbol{h}_t$, then a fully connected layer is utilized to predict the performance on e_{t+1}:

$$\begin{aligned} \widetilde{\boldsymbol{h}}_t &= \boldsymbol{q}_{e_{t+1}} \cdot \boldsymbol{h}_t, \\ y_{t+1} &= \sigma(\boldsymbol{W}_5[\widetilde{\boldsymbol{h}}_t \oplus \boldsymbol{e}_{t+1}] + \boldsymbol{b}_5), \end{aligned} \tag{15}$$

where $\boldsymbol{W}_5 \in \mathbb{R}^{1 \times (d_h+d_e)}$, $\boldsymbol{b}_5 \in \mathbb{R}$ are trainable parameters, y_{t+1} which in the range of (0,1) represents the probability to answer question $\boldsymbol{e}_{t+1}$ correctly. To train all parameters and vectors in LBKT, we choose the cross-entropy log loss between the predicted answer y and actual answer r as the objective function, which will be minimized in the training process:

$$\mathbb{L} = -\sum_{t=1}^{T} (r_t \log y_t + (1 - r_t) \log(1 - y_t)). \tag{16}$$

5 EXPERIMENTS

5.1 Datasets

We evaluate our method on three public datasets: (1) ASSIST2009 [1](2) ASSIST2012[2] and (3) Junyi[3]. The basic statistics of these three datasets are listed in Table 2 and descriptions are as follows:

- **ASSIST2009** is collected from the ASSISTments online tutoring system which records the behaviors of "attempt count", "hint count" and "ms first response". The records without concepts and the learners whose answering sequence is less than ten are

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

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

[3]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198

Table 2: Statistics of all datasets.

Statistics	Datasets		
	ASSIST2009	ASSIST2012	Junyi
Records	297,343	2,622,857	4,316,340
Learners	3,006	22,397	1,000
Questions	9,798	37,413	701
Concepts	107	254	39
Avg.attempts	1.532	1.354	1.417
Avg.time (s)	51.220	54.322	208.398
Avg.hints	0.428	0.394	0.249

removed. To better estimate the distribution of behaviors on each question, we filter out the questions answered less than 10 times.

- **ASSIST2012** is also collected from ASSISTments online tutoring system. We do the same preprocessing as in ASSIST2009.
- **Junyi** is collected from Junyi Academy, a Chinese e-learning platform. Following existing works [50, 51], we select 1000 learners with the most question-answering records from the log data. The data preprocessing is the same as in ASSIST2009.

5.2 Baselines

In order to evaluate the effectiveness of LBKT, we compare it with eight knowledge tracing models. Their details are as follows:

- **DKT** uses RNN to model the question-answer sequence and the hidden state is represented as learners' knowledge state [37]. We use LSTM to implement DKT [21].
- **DKT_concat** is a variant of DKT that we propose to fuse behavioral information, in which features of *Speed*, *Attempts* and *Hints* are concatenated with the original input of DKT.
- **AT-DKT** enhances DKT through adding two auxiliary tasks i.e. question tagging and students' prior knowledge prediction [29].
- **DKVMN** uses a key memory matrix to store the concepts and a dynamic value memory matrix to store and update the learners' proficiency on the corresponding concepts [42].
- **DMKT** estimates learners' knowledge gain from both answering results and learning materials like question-related hints [46]. We feed requested-hints number as learning material to DMKT.
- **SAKT** introduces self-attention mechanism into knowledge tracing to capture the dependency among a learning sequence [36].
- **AKT** applies a monotonic attention mechanism to capture the dependency in a learning sequence and uses the Rasch model to combine questions' individualized difficulty with concepts [17].
- **LPKT** models the learning process of learners by combining their answer time and interval time into calculating the learning gain and forgetting between two continuous time steps [40].

5.3 Training Details

In our experiments, we split the dataset in an 8:1:1 ratio by learners to obtain the training set, validation set, and testing set. For all datasets, we first sorted all learning records of a learner by the timestamp. Then, we set all input sequences to a fixed length as 100. For sequences longer than the fixed length, we cut them into pieces based on the fixed length. For the shorter ones, zero vectors were used to pad them up to the fixed length. We randomly initialized all parameters by the uniform distribution [18]. The parameters d, k, b in Eq. (4) are set to be 10, 0.3, and 0.7 respectively. The rank r in Eq. (12) to break down high-order matrices is set as 4. The dimensions of knowledge state and question embedding are both set as 128 while the answer embedding's dimension is 50 in our implementation. The learning rate is 0.002 with a decay at each epoch and the optimizer is Adam [25]. For fairness, the hyper-parameters of baselines are carefully tuned to have the best performance. All experiments were conducted on a cluster of Linux servers with Tesla V100 GPUs.

5.4 Learner Performance Prediction Results

As it's difficult to directly quantify learners' knowledge states, following existing work [40], we use the performance prediction task to infer how well KT models estimate learners' knowledge states. In order to evaluate the effectiveness of LBKT, we compare it with all baselines on predicting learners' performance and report experimental results in Table 3. We measure the performance of KT models with Root Mean Squared Error (RMSE), Area Under Curve (AUC), and Accuracy (ACC).

In Table 3, we can find several important observations. First, LBKT significantly outperforms all baseline methods on all datasets and evaluation metrics. The superior performance demonstrates that LBKT provides more accurate estimation of learners' knowledge state by capturing learning behaviors' complex effects. Second, DKT_concat which incorporates multiple learning behaviors into DKT promotes its performance and it shows introducing learning behaviors into KT task is necessary and valuable. LPKT which applies behaviors in a finer way outperforms other classic models in almost all metrics demonstrating the significance of digging more into the behaviors' effect. Third, compared with DKT_concat and LPKT that integrate behaviors in a simple way without mining their explicit effect on learners' knowledge state, LBKT achieves better performance which explains the effectiveness of LBKT in capturing the behaviors' distinctive and cooperative effects. We further investigate the efficiency of LBKT in Appendix A.

5.5 Ablation Study

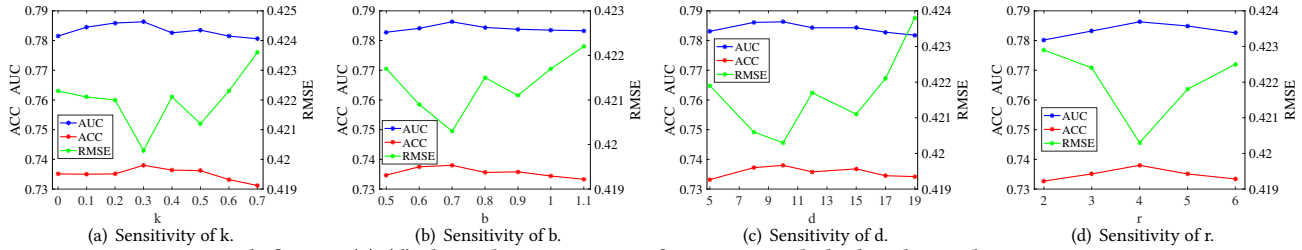
To further investigate the importance of each module in LBKT, we design three variations to conduct the ablation study, each of which removes one part from the original LBKT:

- **LBKT-DB** replaces Differentiated Behavior Effect Quantifying (DBEQ) with a fully connected layer by concatenating behavioral features (i.e. answer time, number of attempts and hints) with previous knowledge state and current interaction embedding.
- **LBKT-FB** replaces Fused Behavior Effect Measuring (FBEM) module with the addition operation to fuse the knowledge acquisition vectors in Eq. (9).
- **LBKT-Forget** refers to LBKT neglecting the learning behaviors' impact in forget gate.

We draw some important conclusions from the results in Table 3. First, the performance of LBKT decreases no matter which part is removed, which proves that all our proposed components are necessary for tracing learners' knowledge state. Second, LBKT-FB brings the biggest decline of performance on ASSIST2009 and ASSIST2012, reflecting that modeling the dependency of multiple behaviors by capturing their complex interaction patterns plays the most important role in LBKT and simply using the addition operation to linearly

Table 3: Results of baselines and ablation experiments on learner performance prediction. Existing state-of-the-art results are underlined and the best results are bold. We compare our LBKT with the SOTA LPKT and * indicates p-value < 0.05 in the t-test. The Bonferroni's correction is used in multiple t-tests.

Methods	ASSIST2009			ASSIST2012			Junyi		
	RMSE	AUC	ACC	RMSE	AUC	ACC	RMSE	AUC	ACC
DKT	0.4372	0.7430	0.7127	0.4226	0.7364	0.7333	0.3536	0.7586	0.8326
DKT_concat	0.4335	0.7508	0.7204	0.4193	0.7447	0.7407	0.3518	0.7655	0.8340
AT-DKT	0.4370	0.7574	0.7172	0.4162	0.7544	0.7440	0.3537	0.7581	0.8325
DKVMN	0.4416	0.7400	0.7038	0.4224	0.7351	0.7359	0.3544	0.7565	0.8324
DMKT	0.4370	0.7569	0.7196	0.4224	0.7377	0.7383	0.3538	0.7586	0.8336
SAKT	0.4545	0.7111	0.6885	0.4236	0.7335	0.7333	0.3544	0.7590	0.8323
AKT	0.4273	0.7766	0.7289	0.4084	0.7760	0.7559	0.3538	0.7593	0.8325
LPKT	0.4236	0.7788	0.7325	0.4078	0.7751	0.7567	0.3509	0.7689	0.8344
LBKT	**0.4203***	**0.7863***	**0.7380***	**0.4043***	**0.7823***	**0.7613***	**0.3494***	**0.7723***	**0.8362***
LBKT-DB	0.4237	0.7793	0.7336	0.4055	0.7815	0.7591	0.3502	0.7698	0.8354
LBKT-FB	0.4243	0.7792	0.7346	0.4056	0.7797	0.7597	0.3496	0.7717	0.8361
LBKT-Forget	0.4212	0.7842	0.7356	0.4054	0.7804	0.7602	0.3494	0.7720	0.8361



(a) Sensitivity of k. (b) Sensitivity of b. (c) Sensitivity of d. (d) Sensitivity of r.

Figure 4: Sub-figures (a)-(d) show the sensitivity of parameters k, b, d and r on dataset ASSIST2009.

combine acquisition vectors under every behavior is insufficient Third, the performance drop of LBKT-DB verifies that compared with implicitly utilizing behaviors as supplementary features, our DBEQ that explicitly quantifies each behavior's distinctive effect on knowledge acquisition is better. Last, introducing behaviors into computing the forget gate assists in capturing the forgetting process and improving the performance. We further present the contribution of different behaviors to LBKT in Appendix B.

5.6 Parameter Sensitivity Analysis

As our main concern is explicitly modeling behaviors' effects on knowledge state, the sensitivity of k, b, and d in Eq. (4), (6) and (8) should be significant. Besides, the fixed rank r in Eq. (12) is also essential to make the decomposition keep as much information of $\boldsymbol{W}_3$ as possible and reduce the computation and storage burden at the same time. To be specific, we evaluate LBKT's performance on eight different numbers of k: {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7}. When applying these different levels on k, d and b always remain 10 and 0.7. Similarly, b varies from {0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1}, d varies from {5, 8, 10, 12, 15, 17, 19}, and r varies from {2, 3, 4, 5, 6}.

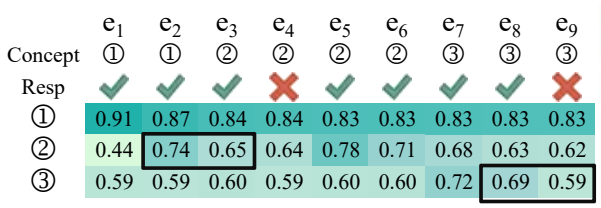
The experimental results on ASSIST2009 are shown in Figure 4(a)-(d). First, for the low bound k, performance increases then decreases when k surpasses 0.3. As a large k will lose the difference of control factor c on different levels of behavior factors (i.e. AC_t, PC_t and NC_t), while a small k enlarges the difference thus leading to large cost on wrongly supposed cases. Second, for threshold b, the performance decreases when it surpasses 0.7 because a small b stands for a lower boundary point for different levels of behavior factors and results in the misunderstanding of behaviors' effect while a large b increases the boundary point. Third, for the discriminative parameter d, performance increases then decreases when d surpasses 10. As the control factor c will change too smoothly between low and high behavior factors on a small d thus losing the discrimination contrasting with the sharp change on a large d. Fourth, for the fixed rank r, LBKT reaches highest performance when r equals to 4. Because a small r will cause more information loss of $\boldsymbol{W}_3$ while a large r introduces more unnecessary information.

5.7 Association Validation Between Learning Behaviors and the Update of Knowledge

We conduct experiments to better investigate the association between knowledge acquisition and behavior factors. We use the Normalized Learning Gain (NL_Gain) [20] to measure the explicit knowledge acquisition of a learner from his performance prediction before and after he answers a question.

$$\begin{aligned} y_t &= \sigma(\boldsymbol{W}_5[\widetilde{\boldsymbol{h}}_{t-1} \oplus \boldsymbol{e}_t] + \boldsymbol{b}_5), \\ y'_t &= \sigma(\boldsymbol{W}_5[\widetilde{\boldsymbol{h}}_t \oplus \boldsymbol{e}_t] + \boldsymbol{b}_5), \\ NL\_Gain &= \frac{y'_t - y_t}{1 - y_t}, \end{aligned} \tag{17}$$

where y_t and y'_t denote the prediction performance before and after the learner answers question e_t. We hold if the learner learns from the question, his performance would increase on the same question. For each behavior factor, we classify records into high and low groups based on the threshold b in Eq. (4) , (6) and (8) which shows best performance when equal to 0.7, and we would get 8 groups according to three behavior factors e.g., group (h, h, l)



	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9
Time	30	25	14	66	61	19	121	6	8
Attempts	1	1	1	2	1	1	1	1	3
Hints	0	0	0	4	0	0	0	0	3
AC	0.76	0.79	0.89	0.36	0.41	0.80	0.04	0.82	0.97
PC	0.31	0.29	0.26	0.52	0.18	0.19	0.16	0.21	0.73
NC	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00	0.99

①Addition and Subtraction Positive Decimals ②Division Fractions ③Addition and Subtraction Fractions

Figure 5: An example of tracking the proficiency of a certain learner on three concepts on the dataset ASSIST2009. The top line represents questions answered by the learner followed by the corresponding concepts and responses below. In addition, the learning behaviors including *Speed*, *Attempts* and *Hints* together with their rescaled factors are presented in the right part.

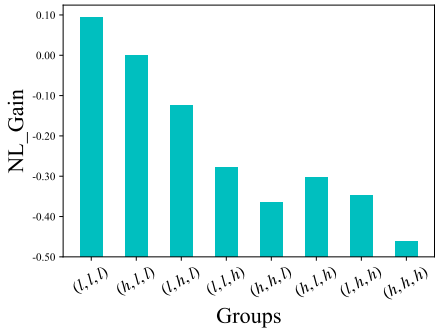


Figure 6: The Normalized Learning Gain of eight groups on ASSIST2009, which are obtained based on the rescaled , e.g., the group (h, h, l) includes records with high *Speed* and *Attempts* factors but a low *Hints* factor.

include records with *h*igh *Speed* and *h*igh *Attempts* factors (i.e. AC_t, PC_t) and a *l*ow *Hints* factor NC_t. For each group, the average NL_Gain of its records represents the knowledge acquisition.

Figure 6 shows the NL_Gain results of each group and we have some observations. First, the NL_Gain of group (l, l, l) is the highest while group (h, h, h) is the lowest. This indicates that knowledge acquisition of groups except (l, l, l) is narrowed by some behavior (i.e. high speed, many attempts and requested hints correlate to poor knowledge acquisition from an overall point of view) and group (h, h, h) is especially affected by all three behaviors. Second, the NL_Gain of groups that infer a high factor on just one behavior e.g., (h, l, l) is higher than those groups that infer high factors on two or three behaviors, which demonstrates the dependency among different behaviors is captured well through FBEM. Third, in the groups that infer a high factor on just one behavior, the NL_Gain of group (l, l, h) is the lowest, which indicates that relying on hints to finish a question influences most on learners' knowledge acquisition.

5.8 Visualization of Proficiency Evolution

A superior ability of LBKT is that it can track learners' knowledge proficiency on each concept as we use a matrix to store the knowledge state under multiple concepts instead of a high-dimensional vector. Learners can intuitively understand their mastery and deficiencies in each knowledge concept. The proficiency of a learner on concept c_m at step t could be computed follows [27]:

$$y_t^m = \sigma(\boldsymbol{W}_5[\boldsymbol{h}_t^m \oplus \mathbf{0}] + \boldsymbol{b}_5), \tag{18}$$

where $\boldsymbol{h}_t^m$ means the row relates to concept c_m in the knowledge state matrix $\boldsymbol{h}_t$ and $\mathbf{0} = (0, 0, ..., 0)$ is the masked question embedding with the same dimension as $\boldsymbol{e}_{t+1}$ in Eq. (15).

Figure 5 shows the changing process of a learner's knowledge proficiency on three concepts traced by LBKT after he finished answering 9 questions sequentially. We find that although the learner has answered question e_3 related to concept "Division Fractions" correctly, his proficiency on it decreases from 0.74 to 0.65. To investigate the reason, we move our sight to his learning behaviors presented in the right part of Figure 5. We note that he spends 14 seconds with one attempt to obtain the correct answer and the *Speed* factor *AC* estimated by DBEQ is extremely high (0.89). This implies the learner answers too fast and is possibly viewed as guessing, so LBKT narrows his knowledge acquisition and together with the forget gate decreases his proficiency. Similarly, when the learner answers question e_9 with high factors as 0.97, 0.73 and 0.99 under three behaviors, the cooperative effect of these three behaviors drives the decline of proficiency in the answering concept "Addition and Subtraction Fractions". LBKT tracks learners' proficiency not only based on their responses but also considers different behaviors' effect, thus making more accurate and reasonable estimation.

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we pointed out the significant effects of learning behaviors on knowledge tracing and proposed LBKT to measure the effects. We first designed the Differentiated Behavior Effect Quantifying (DBEQ) module and Fused Behavior Effect Measuring (FBEM) module to quantify the distinctive and cooperative effects of *Speed*, *Attempts* and *Hints* on knowledge acquisition. Moreover, a forget gate was proposed incorporating the behaviors to capture the decrease of proficiency over time and by combining forgetting and knowledge acquisition the knowledge state was updated. Experimental results on three public datasets showed that LBKT outperformed previous classic KT methods and estimated more reasonable knowledge state of learners considering their behaviors. Our LBKT can introduce other behaviors like the chosen option transitions or answer text submitted by learners when available by analyzing their distributions and effect on knowledge acquisition. In the future, we will try to incorporate more behaviors and deeply mine how these behaviors affect learners' knowledge states.

ACKNOWLEDGMENTS

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2022ZD0117103) and the National Natural Science Foundation of China (Grants No. 62106244, No. U20A20229), the University Synergy Innovation Program of Anhui Province (No. GXXT-2022-042), and the iFLYTEK joint research.

REFERENCES

[1] Solmaz Abdi, Hassan Khosravi, Shazia Sadiq, and Ali Darvishi. 2021. Open learner models for multi-activity educational systems. In *Artificial Intelligence in Education: 22nd International Conference, AIED 2021, Utrecht, The Netherlands, June 14–18, 2021, Proceedings, Part II*. Springer, 11–17.

[2] Vincent Aleven and Kenneth R Koedinger. 2000. Limitations of student control: Do students know when they need help?. In *International conference on intelligent tutoring systems*. Springer, 292–303.

[3] Ashton Anderson, Daniel Huttenlocher, Jon Kleinberg, and Jure Leskovec. 2014. Engaging with massive online courses. In *Proceedings of the 23rd international conference on World wide web*. 687–698.

[4] Ryan Shaun Baker, Albert T Corbett, and Kenneth R Koedinger. 2004. Detecting student misuse of intelligent tutoring systems. In *International conference on intelligent tutoring systems*. Springer, 531–540.

[5] Ryan S.J.d. Baker, Albert T. Corbett, Kenneth R. Koedinger, Shelley Evenson, Ido Roll, Angela Z. Wagner, Meghan Naim, Jay Raspat, Daniel J. Baker, and Joseph E. Beck. 2006. Adapting to when students game an intelligent tutoring system. In *International conference on intelligent tutoring systems*. Springer, 392–401.

[6] Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, and Angela Z Wagner. 2004. Off-task behavior in the cognitive tutor classroom: When students" game the system". In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 383–390.

[7] Ryan Shaun Baker, Ido Roll, Albert T Corbett, and Kenneth R Koedinger. 2005. Do performance goals lead students to game the system?. In *AIED*. 57–64.

[8] Tiffany Barnes. 2005. The Q-matrix method: Mining student response data for knowledge. In *American association for artificial intelligence 2005 educational data mining workshop*. AAAI Press, Pittsburgh, PA, USA, 1–8.

[9] Penghe Chen, Yu Lu, Vincent W Zheng, and Yang Pian. 2018. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 39–48.

[10] Ryan SJ d Baker, Sujith M Gowda, Michael Wixon, Jessica Kalka, Angela Z Wagner, Aatish Salvi, Vincent Aleven, Gail W Kusbit, Jaclyn Ocumpaugh, and Lisa Rossi. 2012. Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. *International Educational Data Mining Society* (2012).

[11] Phung Do, Kha Nguyen, Thanh Nguyen Vu, Tran Nam Dung, and Tuan Dinh Le. 2017. Integrating knowledge-based reasoning algorithms and collaborative filtering into e-learning material recommendation system. In *International Conference on Future Data and Security Engineering*. Springer, 419–432.

[12] John Domingue, Alexander Mikroyannidis, and Stefan Dietze. 2014. Online learning and linked data: lessons learned and best practices. In *Proceedings of the 23rd International Conference on World Wide Web*. 191–192.

[13] Pragya Dwivedi, Vibhor Kant, and Kamal K Bharadwaj. 2018. Learning path recommendation based on modified variable length genetic algorithm. *Education and information technologies* 23, 2 (2018), 819–836.

[14] Susan E Embretson and Steven P Reise. 2013. *Item response theory*. Psychology Press.

[15] Jean-Claude Falmagne, Eric Cosyn, Jean-Paul Doignon, and Nicolas Thiéry. 2006. The assessment of knowledge, in theory and in practice. In *Formal Concept Analysis: 4th International Conference, ICFCA 2006, Dresden, Germany, February 13-17, 2006. Proceedings*. Springer, 61–79.

[16] 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.

[17] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2330–2339.

[18] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[19] José González-Brenes, Yun Huang, and Peter Brusilovsky. 2014. General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge. In *The 7th international conference on educational data mining*. University of Pittsburgh, 84–91.

[20] Richard R Hake. 2002. Relationship of individual student normalized learning gains in mechanics with gender, high-school physics, and pretest scores on mathematics and spatial visualization. In *Physics education research conference*, Vol. 8. 1–14.

[21] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[22] 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.

[23] Jeffrey Johns and Beverly Woolf. 2006. A dynamic mixture model to detect student motivation and proficiency. In *AAAI*. 163–168.

[24] E Joseph. 2005. Engagement tracing: using response times to model student disengagement. *Artificial intelligence in education: Supporting learning through intelligent and socially informed technology* 125 (2005), 88.

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

[26] Jiayu Liu, Zhenya Huang, Chengxiang Zhai, and Qi Liu. 2023. Learning by Applying: A General Framework for Mathematical Reasoning via Enhancing Explicit Knowledge Learning. *arXiv preprint arXiv:2302.05717* (2023).

[27] Qi Liu, Zhenya Huang, Yu Yin, Enhong Chen, Hui Xiong, Yu Su, and Guoping Hu. 2019. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering* 33, 1 (2019), 100–115.

[28] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2021. A survey of knowledge tracing. *arXiv preprint arXiv:2105.15106* (2021).

[29] Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Boyu Gao, Weiqi Luo, and Jian Weng. 2023. Enhancing Deep Knowledge Tracing with Auxiliary Tasks. *arXiv preprint arXiv:2302.07942* (2023).

[30] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. 2018. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064* (2018).

[31] Ting Long, Yunfei Liu, Jian Shen, Weinan Zhang, and Yong Yu. 2021. Tracing knowledge state with individual cognition and acquisition estimation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 173–182.

[32] Kritphong Mongkhonvanit, Klint Kanopka, and David Lang. 2019. Deep knowledge tracing and engagement with moocs. In *Proceedings of the 9th international conference on learning analytics & knowledge*. 340–342.

[33] Kasia Muldner, Winslow Burleson, Brett Van de Sande, and Kurt VanLehn. 2011. An analysis of students' gaming behaviors in an intelligent tutoring system: Predictors and impacts. *User modeling and user-adapted interaction* 21, 1 (2011), 99–135.

[34] Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The world wide web conference*. 3101–3107.

[35] Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *2019 IEEE/WIC/ACM International Conference On Web Intelligence (WI)*. IEEE, 156–163.

[36] Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837* (2019).

[37] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems* 28 (2015).

[38] Shi Pu and Lee Becker. 2022. Self-attention in Knowledge Tracing: Why It Works. In *International Conference on Artificial Intelligence in Education*. Springer, 731–736.

[39] Sarah Schultz and Ivon Arroyo. 2014. Tracing knowledge and engagement in parallel in an intelligent tutoring system. In *Educational Data Mining 2014*.

[40] Shuanghong Shen, Qi Liu, Enhong Chen, Zhenya Huang, Wei Huang, Yu Yin, Yu Su, and Shijin Wang. 2021. Learning process-consistent knowledge tracing. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1452–1460.

[41] Shuanghong Shen, Qi Liu, Enhong Chen, Han Wu, Zhenya Huang, Weihao Zhao, Yu Su, Haiping Ma, and Shijin Wang. 2020. Convolutional knowledge tracing: Modeling individualization in student learning process. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1857–1860.

[42] Xia Sun, Xu Zhao, Bo Li, Yuan Ma, Richard Sutcliffe, and Jun Feng. 2021. Dynamic key-value memory networks with rich features for knowledge tracing. *IEEE Transactions on Cybernetics* (2021).

[43] Yue Suo, Naoki Miyata, Hiroki Morikawa, Toru Ishida, and Yuanchun Shi. 2008. Open smart classroom: Extensible and scalable learning system in smart space using web service technology. *IEEE transactions on knowledge and data engineering* 21, 6 (2008), 814–828.

[44] Nguyen Thai-Nghe, Lucas Drumond, Tomáš Horváth, Artus Krohn-Grimberghe, Alexandros Nanopoulos, and Lars Schmidt-Thieme. 2012. Factorization techniques for predicting student performance. In *Educational recommender systems and technologies: Practices and challenges*. IGI Global, 129–153.

[45] Angel de Vicente and Helen Pain. 2002. Informing the detection of the students' motivational state: an empirical study. In *International Conference on Intelligent Tutoring Systems*. Springer, 933–943.

[46] Chunpai Wang, Siqian Zhao, and Shaghayegh Sahebi. 2021. Learning from Non-Assessed Resources: Deep Multi-Type Knowledge Tracing. *International Educational Data Mining Society* (2021).

[47] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yu Yin, Shijin Wang, and Yu Su. 2022. NeuralCD: A General Framework for Cognitive Diagnosis. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[48] Shuhan Wang, Hao Wu, Ji Hun Kim, and Erik Andersen. 2019. Adaptive learning material recommendation in online language education. In *International conference on artificial intelligence in education*. Springer, 298–302.

[49] Yutao Wang and Neil T Heffernan. 2012. Leveraging First Response Time into the Knowledge Tracing Model. *International Educational Data Mining Society* (2012).
[50] Runze Wu, Guandong Xu, Enhong Chen, Qi Liu, and Wan Ng. 2017. Knowledge or gaming? Cognitive modelling based on multiple-attempt response. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 321–329.
[51] Haiqin Yang and Lap Pong Cheung. 2018. Implicit heterogeneous features embedding in deep knowledge tracing. *Cognitive Computation* 10, 1 (2018), 3–14.
[52] Michael V Yudelson, Kenneth R Koedinger, and Geoffrey J Gordon. 2013. Individualized bayesian knowledge tracing models. In *International conference on artificial intelligence in education*. Springer, 171–180.
[53] Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250* (2017).
[54] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*. 765–774.
[55] Liang Zhang, Xiaolu Xiong, Siyuan Zhao, Anthony Botelho, and Neil T Heffernan. 2017. Incorporating rich features into deep knowledge tracing. In *Proceedings of the fourth (2017) ACM conference on learning@ scale*. 169–172.
[56] Chuang Zhao, Hongke Zhao, Ming He, Jian Zhang, and Jianping Fan. 2023. Cross-domain recommendation via user interest alignment. *arXiv preprint arXiv:2301.11467* (2023).
[57] Siqian Zhao, Chunpai Wang, and Shaghayegh Sahebi. 2020. Modeling knowledge acquisition from multiple learning resource types. *arXiv preprint arXiv:2006.13390* (2020).

A EFFICIENCY OF LBKT

We conduct extensive experiments to show LBKT's efficiency. As the AKT and LPKT models show comparable performance with LBKT while the other baselines show much worse performance, we just compare the training time LBKT consumed with these two baselines. And the results are listed in Table 4.

We adopt early stopping strategy in the training phase if the evaluation metric AUC score on validation set does not increase for 5 epochs. The batch size for training these three models is all set as 32 and the learning rate is tuned to have best performance. We setup the training process for each model five times. And **Time / epoch** refers to the average time (in minutes) consumed each epoch for training. **Best epoch** refers to the average epochs used for the model to achieve best performance on validation set. **Total time** refers to the total time (in minutes) for the model to finish training with the early stopping strategy. We can draw some conclusions from the experimental results. Firstly, LBKT is more efficient than the SOTA LPKT in all the datasets which indicates that LBKT outperforms LPKT not only on the performance but also on the efficiency. Secondly, although LBKT spent more time per epoch than AKT, it converges much faster and thus leading to the comparable consuming time for training with AKT. Besides, LBKT utilizes a matrix to record the learner's proficiency level under each knowledge concept which results in consuming more time each epoch in contrast with AKT that only use a high-dimensional hidden vector to represent the student's proficiency. So our LBKT can obtain better interpretability for exhibiting learner's proficiency under different concepts than AKT. To sum up, it's a good choice to use LBKT in real-time online learning environments if the environment requires good interpretability to show learner's proficiency under each knowledge concept.

B CONTRIBUTION OF DIFFERENT BEHAVIORS

We further conduct ablation experiments of the three behavioral variables to investigate their different contribution. **LBKT_Speed**, **LBKT_Attempt**, and **LBKT_Hint** refer to that modifying LBKT to be kept with only one behavior *Speed*, *Attempt* and *Hint* respectively. **LBKT-none** refers to LBKT with none behaviors which is built by setting response time, attempts number and hints number as fixed number 100 seconds, 1 and 0 respectively. We can draw some conclusions from the experimental results. Firstly, it can be seen that all the three behaviors contributes to the performance of LBKT as **LBKT_Speed**, **LBKT_Attempt** and **LBKT_Hint** all outperform **LBKT_None**. Secondly, the "Speed" behavior contributes most to LBKT on ASSIST2009 and Junyi datasets. Finally, by considering all these three behaviors, LBKT gains great performance improvement.

Table 4: Consuming training time for different models on three widely-used datasets. The best results are bold.

Methods	ASSIST2009			ASSIST2012			Junyi		
	Time / epoch	Best epoch	Total time	Time / epoch	Best epoch	Total time	Time / epoch	Best epoch	Total time
AKT	**0.05**	32.80	**1.89**	**1.20**	26.40	**37.68**	**1.65**	31.20	**59.73**
LPKT	1.03	2.80	8.03	7.14	3.20	58.55	7.31	10.80	115.50
LBKT	0.75	**2.20**	5.40	6.05	**2.00**	42.35	6.73	**5.80**	72.68

Table 5: Results of learners' performance prediction for LBKT fed with different behaviors. The second best results are underlined and the best results are bold.

Methods	ASSIST2009			ASSIST2012			Junyi		
	RMSE	AUC	ACC	RMSE	AUC	ACC	RMSE	AUC	ACC
LBKT	**0.4203**	**0.7863**	**0.7380**	**0.4043**	**0.7823**	**0.7613**	**0.3494**	**0.7723**	**0.8362**
LBKT_None	0.4279	0.7733	0.7276	0.4082	0.7759	0.7549	0.3510	0.7681	0.8341
LBKT_Speed	<u>0.4206</u>	<u>0.7854</u>	<u>0.7373</u>	0.4062	0.7781	0.7583	<u>0.3501</u>	<u>0.7699</u>	<u>0.8356</u>
LBKT_Attempt	0.4224	0.7835	0.7336	<u>0.4058</u>	<u>0.7792</u>	<u>0.7588</u>	0.3506	0.7685	0.8342
LBKT_Hint	0.4223	0.7831	0.7335	0.4066	0.7786	0.7587	0.3503	0.7698	0.8354