Assessing Student's Dynamic Knowledge State by Exploring the Question Difficulty Effect

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

Zhenya Huang
Anhui Province Key Laboratory of Big Data Analysis and Application, 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

Qi Liu
Anhui Province Key Laboratory of Big Data Analysis and Application, 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

Yu Su
School of Computer Science and Technology, Hefei Normal University & Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
Hefei, China
yusu@hfnu.edu.cn

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

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

ABSTRACT

Knowledge Tracing (KT), which aims to assess students' dynamic knowledge states when practicing on various questions, is a fundamental research task for offering intelligent services in online learning systems. Researchers have devoted significant efforts to developing KT models with impressive performance. However, in existing KT methods, the related question difficulty level, which directly affects students' knowledge state in learning, has not been effectively explored and employed. In this paper, we focus on exploring the question difficulty effect on learning to improve student's knowledge state assessment and propose the DIfficulty Matching Knowledge Tracing (DIMKT) model. Specifically, we first explicitly incorporate the difficulty level into the question representation. Then, to establish the relation between students' knowledge state and the question difficulty level during the practice process, we accordingly design an adaptive sequential neural network in three stages: (1) measuring students' subjective feelings of the question difficulty before practice; (2) estimating students' personalized knowledge acquisition while answering questions of different difficulty levels; (3) updating students' knowledge state in varying degrees to match the question difficulty level after practice. Finally, we conduct extensive experiments on real-world datasets, and the results demonstrate that DIMKT outperforms state-of-the-art KT models. Moreover, DIMKT shows superior interpretability by exploring the question difficulty effect when making predictions. Our codes are available at https://github.com/shshen-closer/DIMKT.

*Corresponding Author.

SIGIR '22, July 11–15, 2022, Madrid, Spain.

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531939

CCS CONCEPTS

• **Information systems → Data mining**; • **Social and professional topics → Student assessment**; • **Applied computing →** *Computer-assisted instruction.*

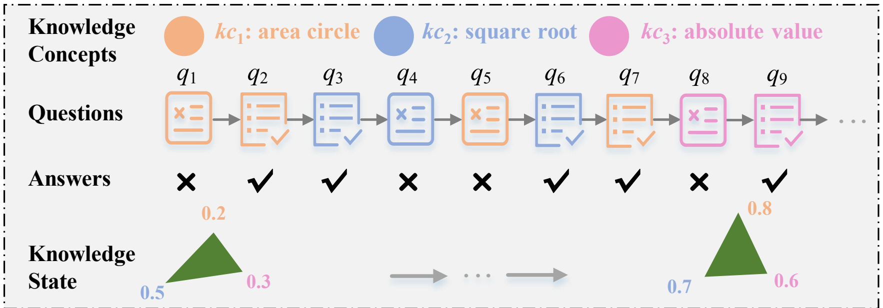
KEYWORDS

data mining, user modeling, knowledge tracing, question difficulty effect, adaptive learning

ACM Reference Format:
Shuanghong Shen, Zhenya Huang, Qi Liu, Yu Su, Shijin Wang, and Enhong Chen. 2022. Assessing Student's Dynamic Knowledge State by Exploring the Question Difficulty Effect. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3477495.3531939

1 INTRODUCTION

Knowledge Tracing (KT) aims to promote teaching and learning in online learning systems [2, 8, 20]. According to students' performance on previous questions, KT measures their knowledge states on different knowledge concepts (e.g., *square root*) and predicts their answers on future questions. Subsequently, based on the knowledge state assessment, we can give valuable feedback to



Questions	Knowledge Concept	Difficulty
q_1	area circle	hard
q_2	area circle	easy
q_5	area circle	hard
q_7	area circle	medium
q_3	square root	medium
q_4	square root	hard
q_6	square root	hard
q_8	absolute value	medium
q_9	absolute value	easy

Figure 1: A toy example of students' practice process and knowledge tracing.

different students about their learning, further tailor adaptive learning schemes for each student [5, 33], and even identify students at risk of failure in the early stage of courses [27]. Benefiting from these intelligent educational services, students can improve their learning efficiency and focus on the poorly mastered knowledge concepts (KCs). Meanwhile, teachers can also instruct students in the light of their knowledge abilities [6, 21].

We give a toy example of KT in the left part of Figure 1, where students practice on different questions to achieve knowledge growth in learning. Considering both questions and their corresponding answers at each practice step, KT models utilize various approaches to capture the evolution of students' knowledge states. For example, Bayesian Knowledge Tracing (BKT) [5] was a particular case of the Hidden Markov Model. Deep Knowledge Tracing (DKT) [36] introduced Recurrent Neural Networks (RNNs) [48] to model students' complex cognitive process. Learning Process-consistent Knowledge Tracing (LPKT) [38] turned to calculate the learning gains and forgetting for better modeling the dynamic learning process.

As shown in the right part of Figure 1, questions are generally distinguished by two basic attributes: KCs and difficulty levels. However, in earlier representative KT models, such as BKT and DKT, questions are only represented by their related KCs without considering their difficulty levels. Therefore, they tend to misestimate the student's knowledge state. For example, in Figure 1, the student got wrong answers on questions q_1 and q_5 with the hard difficulty level, although he/she performed well on questions q_2 and q_7 with the easy and medium difficulty levels, his/her knowledge state on the corresponding KCs (i.e., *area circle*) should only be at the medium level, rather than the high level of 0.8 in Figure 1. Recently, some recent KT models have noticed that exploring the question difficulty factor would be essential for KT. For example, EKT [37] and RKT [31] measured the question difficulty implicitly by analyzing questions' text contents [18]. AKT [10] introduced question embeddings based on Item Response Theory (IRT) [42] to enrich the question representations with their difficulty levels. PEBG [22] and MF-DAKT [52] took advantage of the difficulty level as external information to obtain pre-trained question representations. Although these methods have achieved great success, their effectiveness is limited in only utilizing the question difficulty to improve the question representation.

In fact, the question with different difficulty levels would be a natural indicator to estimate students' knowledge state, i.e., students who can correctly answer harder questions would be regarded to have better knowledge states. However, it is a nontrivial problem to establish the relationship between the student knowledge state and the question difficulty level since the question difficulty effect can produce complicated impacts on student learning. First, students have different knowledge states (even for the same student, his/her knowledge state continues to change in learning), so that they may have quite different subjective feelings about the question difficulty level before practice [7, 17]. There is a common saying in China that *it is easy for one who knows it and difficult for one who does not*, which just reflects this phenomenon. Second, while answering questions with different difficulty levels, students may acquire individual knowledge [24, 32]. For example, students who have already mastered the related KCs can hardly learn anything when practicing on simple questions, while students who are still struggling are expected to make more progress. On the contrary, solving hard questions will be benefit students with better knowledge state, while students with poor mastery will run into a big trouble and even lose learning interests. Third, after finishing the practice, we need to give an objective judgment of students' latest knowledge state, which should be distinctly updated to match the question difficulty level.

In this paper, considering the above question difficulty effects, we aim to improve the KT performance by bridging the relationship between the student's knowledge state and the question difficulty level. To achieve this goal, we propose a novel DIfficulty Matching Knowledge Tracing (DIMKT) model. In DIMKT, we first directly utilize the difficulty level of both the question and the KC to enhance the question representation. Then, we present an Adaptive Sequential Neural Network (ASNN) to realize the connection between the knowledge state and the difficulty level in the practice process. In ASNN, we first capture students' subjective difficulty feelings before practice through calculating the differences between the difficulty-enhanced question embedding and students' knowledge state. Then, we combine students' subjective difficulty feelings and their answers to determine their personalized knowledge acquisition during practice. After students finishing the practice, according to their previous knowledge state, answers, and the question difficulty, we design a knowledge indicator to update their knowledge state with distinctions, which is in line with the question difficulty level. Finally, we conduct extensive experiments on two real-world datasets, the results demonstrate that DIMKT outperforms the state-of-the-art KT methods. Besides, DIMKT also has superior interpretability when making predictions of students' future performance. The main contributions of our paper are summarized as follows:

- We analyze the significant impacts of the question difficulty effect on students' learning in three stages: subjective difficulty feelings before practice, personalized knowledge acquisitions during practice, and distinct knowledge state after practice.
- We present a novel DIfficulty Matching Knowledge Tracing (DIMKT) model to measure the question difficulty effect on learning. In DIMKT, an Adaptive Sequential Neural Network (ASNN) is carefully designed to establish the relationship between the student's knowledge state and the question difficulty level during the practice process.
- Extensive experimental results demonstrate the effectiveness of DIMKT. Besides, DIMKT also has excellent interpretability as it not only predicts what students' answers will be but also gives the reason for the predictions.

2 RELATED WORKS

In this section, we first introduce existing related works about knowledge tracing in detail according to the development timeline, followed by presenting the question difficulty effect.

2.1 Knowledge Tracing

Knowledge tracing has been studied for decades along with the popularity of online learning [38]. The first proposed KT model is BKT, which utilized the Hidden Markov Model to model students' knowledge state [5]. Pardos and Heffernan [34] extended BKT by adding an extra difficulty node. Besides, some factor models, such as Performance Factor Analysis (PFA) [35], took logistic functions to estimate the probability of mastery [4]. Factorization machines (FMs), which were used to encode users and items in recommender systems, were also applied in KT [40, 44]. HawkesKT leveraged the Hawkes process to adaptively model temporal cross-effects in the learning process [45]. In recent years, the powerful ability of non-linearity and feature extraction of neural networks make them well suited to modeling the complex cognitive process of students. Specifically, DKT firstly introduced deep learning into KT [36], which utilized RNNs or LSTMs [13, 48] to model the students' knowledge state. Then, DKVMN used memory networks to store the latent KCs and update students' related knowledge proficiency [51]. GKT proposed to use graph neural networks (GNNs) [49] to model the naturally existing graph structure within the KCs [49]. Similarly, SKT also used GNNs to capture the influence propagation among KCs in KT [41]. CKT applied Convolutional Neural Networks (CNNs) [19] to model students' individualized learning rates [39]. PEBG presented a method to obtain pre-trained question embeddings for KT [22]. MF-DAKT also introduced a pre-training method to incorporate the question relation and difficulty into question representations, which further applied a dual-attentional mechanism for conducting KT [52]. Although PEBG and MF-DAKT developed different pre-train methods, they both obtained pre-trained question representations from their different difficulty levels. EKT leveraged the effectiveness of questions' text contents to enhance the performance of KT [37]. SAKT introduced the self-attention mechanism in Transformer [43] to the KT task [30]. RKT then utilized the contextual information of questions to enhance the self-attention mechanism for KT [31]. In EKT and RKT, the question difficulty level could be seen as a side product in questions' text contents. AKT proposed to utilize the IRT model in psychometrics [25] to construct embeddings for questions and KCs, which further incorporated the self-attention mechanism with monotonic assumption and utilized the encoder-decoder architecture in KT [10]. It is worth noting that AKT introduced question difficulty to enrich the question embeddings by IRT model-based embeddings [42]. IEKT considered students' individual cognition level and knowledge acquisition sensitivity and estimated them explicitly [24]. LPKT presented a novel paradigm for KT by modeling students' learning gains and forgetting in continuous timesteps rather than learning outcomes in single timesteps [38].

2.2 The Question Difficulty Effect

Generally, there is a close link between the question difficulty and students's knowledge state it reflects, which has been testified by many previous studies. For example, Knäuper et al. [17] found that students with higher knowledge ability are more able than those with lower ability to provide accurate answers when responding to tough questions. Besides, students with better knowledge state were less impacted by the fluctuation of the question difficulty. Lomas et al. [23] indicated that easy questions could bring more engagement but slower rates of learning, while more challenging questions may lead to faster learning. Beside, from the perspective of question creation, the question difficulty is also of great significance: Too easy or hard questions are unable to distinguish students' different knowledge state [29]. Therefore, many studies have been proposed to efficiently and automatically measure the question difficulty [12, 14]. In some psychological theories, such as the Classic Test Theory (CTT) [1], the question difficulty was calculated from a statistical point of view. Specifically, it defined the question difficulty as the proportion of a well-defined group of students that answered a question correctly. Besides, Item Response Theory (IRT) thought that question difficulty and student ability depended probabilistically on students' answers and measured them by a logistic function [26]. Moreover, the question's difficulty positively connects to the KCs it contains. Beck et al. [3] pointed out that the harder the question, the more KCs required to solve it. In addition, the knowledge itself may put certain constraints on the question, which influence its difficulty. Therefore, these knowledge constraints must be considered when determining question difficulty. Hwang [15] directly regarded that the difficulty level of questions only depended on the number of KCs to be learned.

In summary, the question difficulty significantly affects students' learning, and students' answers to questions of different difficulty levels directly reflect their knowledge state. Some existing KT models have attempted to incorporate the question difficulty level into KT. Nevertheless, they have certain limitations in indirect utilization of question difficulty level, i.e., they implicitly combine the difficulty information into the question representation. We note that some IRT-related models have considered the relations of question difficulty and student ability by logistic functions [46]. However, they assume that the student's knowledge ability is stable in test scenarios, which is not suitable for tracking the student's dynamic knowledge state in daily practice.

In our work, we mainly consider the question difficulty effect on student learning, not only introducing the question difficulty into its representation. Moreover, we set up the relationship between student's knowledge state and the question difficulty level for better assessing students' dynamic knowledge state.

3 PRELIMINARY

In this section, we give a formal definition of knowledge tracing and the question difficulty. Besides, we present some important embeddings in DIMKT. The mathematical notations utilized throughout our paper are summarized in Table 1.

3.1 Problem Definition

In an online intelligent learning system, supposing that the student set is $\mathbb{S} = \{s_1, s_2, ..., s_I\}$ with I different students, the question set is $\mathbb{Q} = \{q_1, q_2, ..., q_J\}$ with J unique questions, and the KC set is $\mathbb{K} = \{k_1, k_2, ..., k_M\}$ with M various KCs. Each question is related to specific KCs, and every student practices on different questions to achieve knowledge mastery. During the learning process, a student's practice sequence is denoted as $X = \{(q_1, a_1), (q_2, a_2), ..., (q_T, a_T)\}$, where q_t is the question answered at time step t, a_t is the related answer correctness label (i.e., 1 for correct answers and 0 for incorrect answers), T is the length of the learning sequence. Knowledge tracing is defined as following:

Definition 3.1. **(Knowledge Tracing).** Given students' historical practice sequence $X = \{(q_1, a_1), (q_2, a_2), ..., (q_T, a_T)\}$, the KT task aims to assess students' evolving knowledge state in learning and predict performance on future questions, which can be further applied to individualize students' learning schemes and maximize their learning efficiency.

3.2 Question Difficulty Definition

As the KCs contained in a question also have distinctive difficulties (e.g., *Multiplication and Division Decimals* is harder than *Addition and Subtraction Decimals*) [3, 15], we define the question in two levels: Question Specific (QS) difficulty and Knowledge Concept (KC) difficulty. Inspired by the CTT and some previous works [14, 28, 52], we use an objective statistical means to calculate the QS and KC difficulty as follows:

$$\begin{aligned} QS &= \sum_{i}^{|S_i|} \frac{\{a_{ij} == 1\}}{|S_i|} \cdot C_{qs}, \\ KC &= \sum_{i}^{|S_i|} \frac{\{a_{im} == 1\}}{|S_i|} \cdot C_{kc}, \end{aligned} \tag{1}$$

where S_i is the set of students who answerws the question q_j or the KC kc_m, $a_{ij} == 1$ and $a_{im} == 1$ refers to corresponding correct answers respectively, and the constant C_{qs} and C_{kc} are the predefined levels of QS and KC difficulty (e.g., there are 100 different QS difficulty levels if C_{qs} is set as 100).

Definition 3.2. **(Question Difficulty).** The question difficulty contains both question specific difficulty QS and knowledge concept difficulty KC, where the former is computed by the proportion of students that answer a question correctly, and the latter is defined as the proportion of students that answer questions containing a knowledge concept correctly.

Notations	Descriptions
$\mathbb{S}, \mathbb{Q}, \mathbb{K}$	The set of student, question, and knowledge concept.
I, J, M	The number of student, question, and knowledge concept.
X	Students' learning sequence.
$\boldsymbol{h}$	Students' knowledge state.
$\boldsymbol{x}$	Difficulty-enhanced question embedding.
$k_m, \boldsymbol{k_m}$	The knowledge concept and its embedding.
$q, \boldsymbol{q}$	The question and its embedding.
$a, \boldsymbol{a}$	Students' actual answer and its embedding.
$QS, \boldsymbol{QS}$	Question specific difficulty and its embedding.
$KC, \boldsymbol{KC}$	Knowledge concept difficulty and its embedding.
C_{qs}, C_{kc}	Constants of QS and KC difficulty levels.
$\boldsymbol{SDF}$	Subjective difficulty feeling.
$\boldsymbol{PKA}$	Personalized knowledge acquisition.
y	Prediction of students' performance.

Table 1: Mathematical notations and descriptions.

3.3 Embeddings

To better understand the whole structure of DIMKT before presenting its details, we give a simple introduction to the essential embeddings in DIMKT from three categories as below.

3.3.1 Difficulty Embedding. Difficulty embedding is the embedding of question difficulty. As the question difficulty contains both QS and KC difficulty, the difficulty embedding is also composed of QS and KC embedding. More specifically, due to we have defined C_{qs} different difficulty levels for QS, we represent the QS embedding by an embedding matrix $\boldsymbol{QS} \in \mathbb{R}^{C_{qs} \times d_{qs}}$, where d_{qs} is the dimension. Therefore, for question q_j with a specific difficulty level C_{qs}^{j} of QS, we can directly obtain its QS embedding from $\boldsymbol{QS}$. Similarly, we use an embedding matrix $\boldsymbol{KC} \in \mathbb{R}^{C_{kc} \times d_{kc}}$ (d_{kc} is the dimension) to represent the KC embedding. We can also get the KC embedding of a question according to its difficulty level of KC.

3.3.2 Knowledge State Embedding. Knowledge state embedding refers to the embedding of students' knowledge state, which is dynamic throughout the learning process. In DIMKT, we use the vector $\boldsymbol{h_t} \in \mathbb{R}^{d_k}$ to represent students' knowledge state at time step t in DIMKT, where d_k is the dimension. It is worth noting that some existing KT models utilize the knowledge matrix to store and update students' knowledge state [38, 51]. Nevertheless, there are complex relations between different KCs, which are hard to distinguish [41]. For example, *one digit division* is the prerequisite of *two digit division*, while they are also similar KCs. Besides, manually-labeled KCs of questions may have inevitable errors and subjective bias [46]. Therefore, we choose to use vector $\boldsymbol{h}$ to store and update students' knowledge state in DIMKT. The experimental results also indicate the vector $\boldsymbol{h}$ is capable of representing the knowledge state as DIMKT achieves better performance than existing best methods.

3.3.3 Question, Answer, and KC Embeddings. In addition to the question difficulty and knowledge state, questions, answers, and KCs are also necessary elements in DIMKT. Specifically, we use an embedding matrix $\boldsymbol{q} \in \mathbb{R}^{J \times d_q}$ (d_q is the dimension) to represent all questions in DIMKT. Besides, another embedding matrix $\boldsymbol{a} \in$

$\mathbb{R}^{2\times d_a}$ (d_a is the dimension), which is made up of two vectors (one for correct and another for incorrect), is utilized to represent students' answers. Besides, if the question is related to k_m, then its KC embedding will be the one-hot vector $\boldsymbol{k}_m \in \mathbb{R}^M$, where the m-th element is 1 and others are 0.

4 THE DIMKT MODEL

In this section, we present the DIMKT model in detail. The main structure of DIMKT is depicted in Figure 2. In DIMKT, we first use both *QS* and *KC* difficulty to enhance the question representations based on the question and KC embeddings. We then design an Adaptive Sequential Neural Network (ASNN) to establish the relationship between student knowledge state and the question difficulty level in the practice process. ASNN can adaptively capture students' knowledge states in a sequential manner by connecting the question difficulty level respectively before, during, and after the practice process. Finally, DIMKT takes advantage of the inner product of students' knowledge state and the question embedding to model the process that students apply their knowledge to answer questions. Their performance can also be predicted by this means.

4.1 Difficulty-enhanced Question Embedding

As mentioned above, earlier representative KT models represented the question by its related KCs without considering the question difficulty level. Such oversimplification is necessary at the early research stage, avoiding the sparsity problem when the number of questions is large. However, it prevents the further development of KT as the question difficulty is of great significance when students answer questions. Therefore, some recent KT models have incorporated question difficulty into the question embedding by different manners [10, 22, 37, 52]. Nevertheless, they only used the question difficulty information implicitly to get more complete question representations. Besides, the relative *KC* difficulty is omitted, which is critical and necessary auxiliary information for the *QS* difficulty. In DIMKT, we take advantage of both the *QS* and *KC* difficulty to enhance the question representation. To be specific, we directly combine the original question embedding $\boldsymbol{q}_t$ and KC embedding $\boldsymbol{k}_m$ with the *QS* difficulty $\boldsymbol{QS}_t$ and the *KC* difficulty $\boldsymbol{KC}_t$ together, followed by a multi-layer perception (MLP) to output the difficulty-enhanced question embedding $\boldsymbol{x}_t$. The specific calculation formula is as follows:

$$\boldsymbol{x}_t = \boldsymbol{W}_1^T[\boldsymbol{q}_t \oplus \boldsymbol{k}_m \oplus \boldsymbol{QS}_t \oplus \boldsymbol{KC}_t] + \boldsymbol{b}_1, \tag{2}$$

where $\oplus$ is the concatenation operation, $\boldsymbol{W}_1 \in \mathbb{R}^{(d_q+M+d_{qs}+d_{kc})\times d_k}$ is the weight matrix, $\boldsymbol{b}_1 \in \mathbb{R}^{d_k}$ is the bias term, d_k is the dimension.

4.2 Adaptive Sequential Neural Work

After getting the difficulty-enhanced question embedding, we need to further bridge the relationship between the student knowledge state and the question difficulty level in the practice process. Intuitively, question difficulty is an inherent attribute of the question, so we can directly combine the question difficulty to enhance the question representation. In contrast, the impacts of the question difficulty effect on the practice process is dynamic, which is much more challenging to be utilized for knowledge state assessment. Concretely, such influence mainly exists in three stages: Firstly, before answering a question, students' subjective feelings of its difficulty vary from person to person. For example, if asking a student who never knows the KC *absolute value* to answer the easy question q_9 in Figure 1, he/she will feel that q_9 is too hard to answer. However, after enough practicing, he/she may even think that the question q_8 of a hard difficulty level is easy, due to he/she has mastered *absolute value* very well. Secondly, during the answering process, students' knowledge acquisition is also personalized. For example, if a student feels that a question is difficult, he/she will benefit a lot by working hard to solve it, while he/she may learn nothing when the question is considered as no challenge. Thirdly, after completing the practice, students' knowledge state should be updated to varying degrees. In other words, we should improve the student's knowledge state if he/she successfully answers hard questions. Meanwhile, students who got wrong answers on easy questions should be assigned lower knowledge state.

In order to model the above complicated influence of the question difficulty effect, we design an Adaptive Sequential Neural Work (ASNN) to set up the relationship between the student knowledge state and the question difficulty level in the learning process. Accordingly, ASNN contains three components as shown in Figure 2: (1) subjective difficulty feeling, (2) personalized knowledge acquisition, and (3) knowledge state updating.

Subjective difficulty feeling. To calculate students' subjective feelings about the question difficulty, we consider the difference between the difficulty-enhanced question embedding $\boldsymbol{x}_t$ and students' previous knowledge state $\boldsymbol{h}_{t-1}$, i.e., students will feel difficult if their knowledge state cannot meet the requirement of the question. Therefore we can get the subjective feeling as follows:

$$\begin{aligned}
\widetilde{\boldsymbol{SDF}}_t &= tanh(\boldsymbol{W}_2^T(\boldsymbol{x}_t - \boldsymbol{h}_{t-1}) + \boldsymbol{b}_2), \\
\boldsymbol{\Gamma}_t^{SDF} &= \sigma(\boldsymbol{W}_3^T(\boldsymbol{x}_t - \boldsymbol{h}_{t-1}) + \boldsymbol{b}_3), \\
\boldsymbol{SDF}_t &= \boldsymbol{\Gamma}_t^{SDF} \cdot \widetilde{\boldsymbol{SDF}}_t,
\end{aligned} \tag{3}$$

where *tanh* is the non-linear activation function, σ is the sigmoid activation function, $\boldsymbol{W}_2, \boldsymbol{W}_3 \in \mathbb{R}^{(d_k)\times d_{sdf}}$ are the weight matrices, $\boldsymbol{b}_2, \boldsymbol{b}_3 \in \mathbb{R}^{d_{sdf}}$ are the bias terms. Here, $\widetilde{\boldsymbol{SDF}}_t$ is the direct output of subjective feeling, which contains difficulty cognition under specific knowledge state. Based on $\widetilde{\boldsymbol{SDF}}_t$, we further design a gate $\boldsymbol{\Gamma}_t^{SDF}$ to choose and reserve the important features in $\widetilde{\boldsymbol{SDF}}_t$. Then, we can get students' subjective difficulty feelings $\boldsymbol{SDF}_t$.

Personalized knowledge acquisition. Students' personalized knowledge acquisition is closely related to their subjective feelings of the question difficulty $\boldsymbol{SDF}_t$ and their answers $\boldsymbol{a}_t$. Generally speaking, the student who feels the question is hard but finally solves it should have more gains. Therefore, we incorporate $\boldsymbol{SDF}_t$ and $\boldsymbol{a}_t$ together to get the knowledge acquisition in a similar way to the subjective feeling as follows:

$$\begin{aligned}
\widetilde{\boldsymbol{PKA}}_t &= tanh(\boldsymbol{W}_4^T(\boldsymbol{SDF}_t \oplus \boldsymbol{a}_t) + \boldsymbol{b}_4), \\
\boldsymbol{\Gamma}_t^{PKA} &= \sigma(\boldsymbol{W}_5^T(\boldsymbol{SDF}_t \oplus \boldsymbol{a}_t) + \boldsymbol{b}_5), \\
\boldsymbol{PKA}_t &= \boldsymbol{\Gamma}_t^{PKA} \cdot \widetilde{\boldsymbol{PKA}}_t,
\end{aligned} \tag{4}$$

where $\boldsymbol{W}_4, \boldsymbol{W}_5 \in \mathbb{R}^{(d_{sdf}+d_a)\times d_k}$ are the weight matrices, $\boldsymbol{b}_4, \boldsymbol{b}_5 \in \mathbb{R}^{d_k}$ are the bias terms. Here $\boldsymbol{\Gamma}_t^{PKA}$ is the gate that control the output

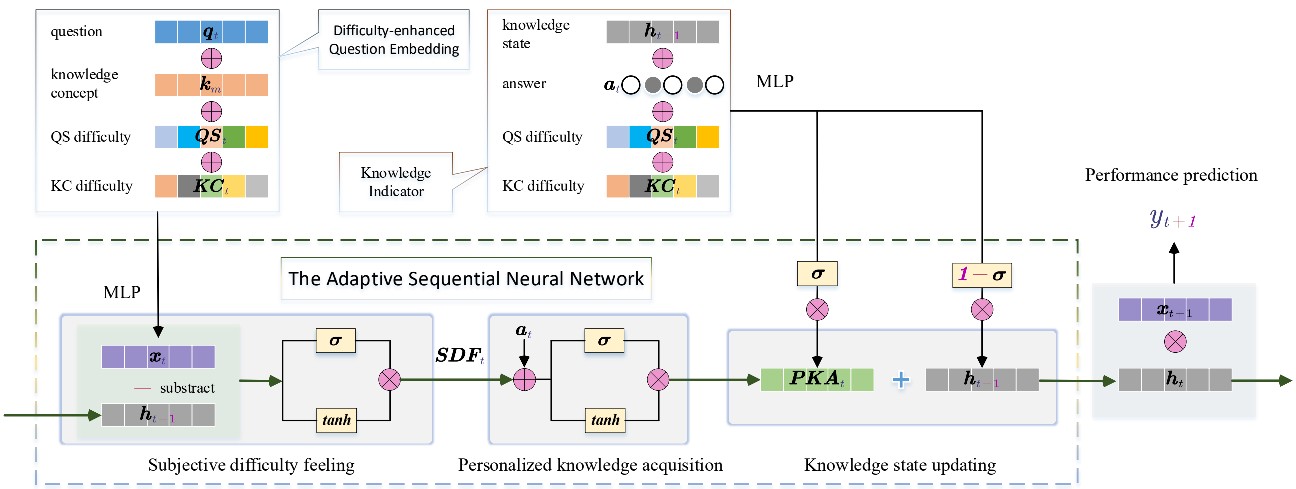


Figure 2: The main structure of our DIMKT model. We show the processing pipeline of DIMKT at timestep t. In this timestep, the input is the question embedding q_t, the knowledge concept k_m of the question, the question specific difficulty QS_t, the knowledge concept difficulty KC_t, the student's answer a_t and previous knowledge state h_{t-1}. The output is the student's updated knowledge state h_t. Moreover, we can also predict the student's performance y_{t+1} at timestep $t+1$.

information in $\widetilde{PKA}_t$. Then, we can obtain students' individual knowledge acquisition during practice.

Knowledge state updating. After practice, we need to objectively update students' knowledge state to varying degrees. For example, if a student successfully solved harder questions, we should improve his/her knowledge state to a higher level. To achieve such updating, we mainly consider three factors: students' previous knowledge state, answers, and the question difficulty. Specifically, we present a knowledge indicator $\Gamma_t^{KSU} \in \mathbb{R}^{d_k}$ according to the above three factors, and then the knowledge indicator is applied to decide the student's updated knowledge state as follows:

$$\begin{aligned} \Gamma_t^{KSU} &= \sigma(\boldsymbol{W}_6^T(\boldsymbol{h}_{t-1} \oplus \boldsymbol{a}_t \oplus \boldsymbol{QS}_t \oplus \boldsymbol{KC}_t) + \boldsymbol{b}_6), \\ \boldsymbol{h}_t &= \Gamma_t^{KSU} \cdot \boldsymbol{h}_{t-1} + (1 - \Gamma_t^{KSU}) \cdot \boldsymbol{PKA}_t, \end{aligned} \tag{5}$$

where $\boldsymbol{W}_6 \in \mathbb{R}^{(d_{qs}+d_{kc}+d_k+d_a)\times d_k}$ is the weight matrix, $\boldsymbol{b}_6 \in \mathbb{R}^{d_k}$ is the bias term. Therefore, both students' previous knowledge state and present knowledge acquisition will affect their latest knowledge state, and the knowledge indicator utilizes questions' difficulty levels to make a trade-off between them for updating students' knowledge state.

4.3 Prediction and Objective Function

As the student's knowledge state $\boldsymbol{h}_t$ at timestep t has been obtained, we can further utilize $\boldsymbol{h}_t$ to predict students' future performance at timestep $t+1$. In DIMKT, we utilize the inner product of the knowledge state vector $\boldsymbol{h}_t$ at timestep t and difficulty-enhance question embedding $\boldsymbol{q}_{t+1}$ at timestep $t+1$ to simulate the practice process that students apply their learned knowledge to answer questions. Then, the probability of correct answers can be inferred from the inner product results by a σ function:

$$y_{t+1} = \sigma(\sum(\boldsymbol{h}_t \cdot \boldsymbol{q}_{t+1})). \tag{6}$$

To train all parameters and vectors in DIMKT, we choose the cross-entropy log loss between the predicted answer y and actual answer a as the objective function, which will be minimized using Adam optimizer [16] on mini-batches, as follows:

$$\mathbb{L} = -\sum_{t=1}^{T}(a_t \log y_t + (1-a_t)\log(1-y_t)) + \lambda_\theta ||\theta||^2. \tag{7}$$

where θ denotes all trainable parameters and embeddings of DIMKT and λ_θ represents the regularization hyperparameter.

5 EXPERIMENTS

In this section, we first introduce the real-world datasets used in our experiments, followed by describing the training details and baseline models. Subsequently, we present the results of all comparison methods on student performance prediction. Moreover, we conduct several experiments to show the interpretability of DIMKT from the following aspects: (1) how various predefined difficulty levels (i.e., C_{qs} and C_{kc}) will affect the performance of DIMKT; (2) the influence of each individual element in DIMKT; (3) the interpretability of the learned difficulty-enhanced question embedding; (4) DIMKT gives the reason for student performance prediction.

5.1 Datasets

We choose two real-world public datasets with different sizes to evaluate the effectiveness of DIMKT:(1) ASSIST2012[1], and (2) Eedi2020[2]. The statistics of the datasets are listed in Table 2. The detailed descriptions of the datasets are as follows:

- **ASSIST2012** is a dataset collected from an online tutoring system, i.e., ASSISTments [9]. This dataset is gathered from the skill

[1] https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect
[2] https://eedi.com/projects/neurips-education-challenge

Statistics	Datasets	
	ASSIST2012	Eedi2020
# of practice records	2,711,813	15,867,849
# of students	29,018	118,971
# of questions	53,091	27,613
# of KCs	265	282
Avg.records per student	93.45	133.38
Avg.records per question	51.08	574.65
Avg.records per KC	10,233.26	56,268.97

Table 2: Statistics of all datasets.

builder problem sets, where students need to practice on similar questions for obtaining knowledge. In this dataset, we have filtered the learning records that the related KCs are missing.

- **Eedi2020** is published in the NeurIPS 2020 Education Challenge, which provided data from two school years of students' answers to mathematics questions from an educational platform, i.e., Eedi [47]. This dataset has hierarchical KCs, where the KC in the lower level is contained in the upper level. We utilize only the most fine-grained KC for each question in our experiments.

5.2 Experimental Setup

In our experiments, we first sorted all practice records of the student by the timestamp of answering. Then, we set all input sequences to a fixed length of 100 based on the average sequence length of the dataset. For sequences longer than the fixed length, we cut them into several unique sub-sequences based on the fixed length. Zero vectors were used to pad the sequences up to the fixed length for the sequences shorter than the fixed length. Since the difficulty value calculated in the statistic means may be unreliable when only a few students answer the question, we dropped the questions with less than 30 answer records. To set up the training process, we randomly initialized all parameters in the uniform distribution [11]. For all datasets, we performed standard 5-fold cross-validation for all models. Specifically, for each fold, 80% of the learning sequences were split as the training set and validation set (their ratio is 8:2), the rest 20% were used as the testing set. All the hyper-parameters are learned on the training set, and the model that performed best on the validation set was used to evaluate the testing set.

The difficulty levels of the QS and KC difficulty are both 100, i.e., C_{qs} and C_{kc} are set to 100. For convenience, the parameters d_{qs}, d_{kc}, d_a, d_k, d_{sdf} are all set to be 128 in our implementation of DIMKT. Other carefully selected parameters may bring slightly better performance, we do not pay time to find better ones, as it is not our main concern in this paper. The learning rate is 0.002, and we set the learning rate decay of 50% every five epochs to achieve the optimal point.

5.3 Baselines

In order to evaluate the effectiveness of DIMKT, we compare it with nine different baselines. All these methods are tuned to have the best performances for a fair comparison. All models are trained on a cluster of Linux servers with the NVIDIA RTX 3090 GPU. The details of all baselines are as follows:

- **DKT** is the first proposed deep learning-based KT model, which leverages RNNs/LSTMs to assess students' knowledge state [36]. We utilized LSTMs to realize DKT in our implementation.
- **DKT+** finds there are two problems in DKT [50]: First, DKT fails to reconstruct the observed input learning sequence. Second, the prediction of DKT on students' future performance across time-steps is not consistent. Therefore, it introduces some regularization terms to solve these two problems.
- **DKVMN** is a KT model based on memory networks [51]. It defines a *key* matrix to store latent KCs and a *value* matrix to store students' knowledge state. Moreover, it propose *read* and *write* operations to update students' knowledge state over time.
- **SAKT** directly applies the transformer to the KT task [30]. It proposes a self-attentive model to capture long-term dependencies between students' learning records.
- **CKT** focuses on the student-specific characteristics in KT [39], it utilizes statistical means to capture students individualized prior knowledge and introduces convolutional windows in CNNs to model their different learning rates in the learning process.
- **PEBG** uses the question difficulty and other information to obtain pre-trained question embeddings for improving the performance of KT methods [22]. Our experiments utilize DKT and the pre-trained question embeddings to realize PEBG.
- **EKT** introduces the text contents of questions into KT [37]. It is worth noting that the question difficulty contained in the text is also included in the question embedding of EKT. In our experiments, we implemented EKT with an attention mechanism.
- **AKT** is the context-aware attentive knowledge tracing model [10]. It defines a knowledge retriever to capture students' dynamic knowledge state by attention mechanism. The question difficulty is utilized to improve the question embeddings by IRT, and two self-attentive encoders are designed to learn context-aware representations of the questions and answers.
- **LPKT** is the learning process-consistent knowledge tracing model [38]. Unlike other KT models that measure students' learning outcomes in single time points, it designs a specific architecture to model students' learning process and calculates students' learning gains and forgetting in continuous time points to update their knowledge state.

5.4 Students' Future Performance Prediction

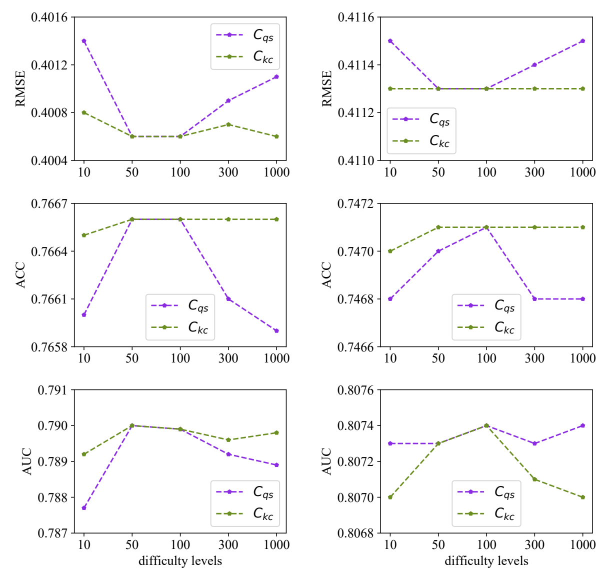
Generally, higher accuracy on students' future performance prediction stands for better estimations of their knowledge state. In order to evaluate the effectiveness of DIMKT, we compare it with all baselines on this task. Specifically, we get the student's practice records from timestep 1 to T, the future performance prediction task aims to predict student performance at each timestep $t, (1 < t <= T)$ based on their performance at timestep 1 to $t-1$.

For providing robust evaluation results, we utilize three evaluation metrics in all experiments, i.e., (1) Root Mean Squared Error (RMSE), (2) Accuracy (ACC), and (3) Area Under Curve (AUC). We set a threshold of 0.5 when calculating the accuracy. Table 3

Datasets	Metrics	DKT [36]	DKT+ [50]	DKVMN [51]	SAKT [30]	CKT [39]	PEBG [22]	EKT [37]	AKT [10]	LPKT [38]	DIMKT (ours)
ASSIST2012	RMSE	0.4226	0.4291	0.4243	0.4235	0.4213	0.4127	0.4070	0.4038	<u>0.4037</u>	**0.4006***
	ACC	0.7375	0.7266	0.7353	0.7364	0.7391	0.7529	0.7594	<u>0.7638</u>	0.7634	**0.7666***
	AUC	0.7304	0.7119	0.7255	0.7279	0.7351	0.7652	0.7732	<u>0.7841</u>	0.7824	**0.7899***
Eedi2020	RMSE	0.4298	0.4422	0.4326	0.4311	0.4302	0.4152	0.4137	<u>0.4128</u>	0.4133	**0.4113***
	ACC	0.7202	0.6974	0.7157	0.7180	0.7193	0.7418	0.7435	<u>0.7453</u>	0.7446	**0.7471***
	AUC	0.7649	0.7337	0.7576	0.7620	0.7644	0.8005	0.8024	0.8041	<u>0.8042</u>	**0.8074***

Table 3: Results of all comparison methods on the student performance prediction task. Existing state-of-the-art results are marked by the underline and the best results are bold. * indicates p-value < 0.05 in the t-test.

gives the experimental results (i.e., the average results of 5-fold cross-validation on the testing set) and the statistical significance of our model against the best baseline model. In Table 3, we can find several important observations. Firstly, DIMKT outperforms all baseline methods on all datasets and evaluation metrics. The superior performance of DIMKT demonstrates that considering the question difficulty effect on student learning is necessary and valuable. Secondly, existing best KT models (e.g., PEBG, EKT, and AKT) that utilize the question difficulty to enhance question embeddings outperform earlier KT models (e.g., DKT, DKVMN, CKT) by a large margin, which also proves the necessity of the question difficulty. Thirdly, in contrast to applying the question difficulty in question embeddings, DIMKT further considers the question difficulty effect in the practice process and successfully establish the relationship between student knowledge state and question difficulty level, therefore it achieves better performance. Finally, compared with LPKT, which directly models students' learning process, DIMKT also gets the better performance, inspiring us to combine the advantage of LPKT and DIMKT for achieving more success in the future.



(a) Results on ASSIST2012. (b) Results on Eedi2020.

Figure 3: The performance of DIMKT under different predefined difficulty levels.

5.5 Sensitivity Analysis of Difficulty Levels

In above experiments, the difficulty levels C_{qs} and C_{kc} are set to 100. As our main concern in this paper is the question difficulty effect on student learning, the sensitivity of difficulty levels should be essential. Therefore, we evaluate the effects of different difficulty levels on the performance of DIMKT. To be specific, we evaluate DIMKT's performance on five different difficulty levels: 10, 50, 100, 300, and 1000. When applying these different difficulty levels on the *QS* difficulty, C_{kc} is always set as 100. Conversely, C_{qs} also remains 100 when C_{kc} takes different values. The experimental results on all datasets are shown in Figure 3, where we can get many reasonable findings. Firstly, for the *QS* difficulty, the specificity of question difficulty is lost when the difficulty level is small, while the generality of question difficulty declines when the difficulty level is high. Therefore, DIMKT has certain degrees of performance degradation when the difficulty level is very small (i.e., 10) or large (i.e., 1000). Based on our experiments, the best difficulty level should strike a balance between specificity and generality, which is between 50 and 100. Secondly, we utilize the *KC* difficulty as auxiliary information in the question difficulty. Therefore, different C_{kc} should have almost no influence on DIMKT's performance, which is also reflected by the horizontal orange line in Figure 3.

5.6 Ablation Study

In this part, we further conduct the ablation study to show the influence of each individual element in DIMKT. We totally choose five variants of DIMKT, each of which removes one element from the original DIMKT. The details of them are as follows:

- DIMKT w/o *KC*, which refers to DIMKT without considering the *KC* difficulty.
- DIMKT w/o *QS*, which refers to DIMKT without considering the *QS* difficulty.
- DIMKT w/o Difficulty, which refers to DIMKT without considering both the *KC* and *QS* difficulty.
- DIMKT w/o ***SDF***, which refers to DIMKT without students' subjective difficulty feelings module in ASNN.
- DIMKT w/o ***PKA***, which refers to DIMKT without students' personalized knowledge acquisition module in ASNN.

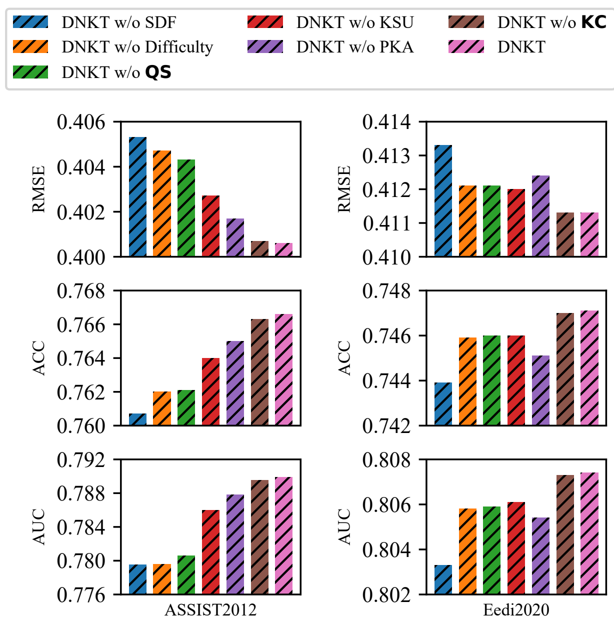


Figure 4: The reuslts of ablation study.

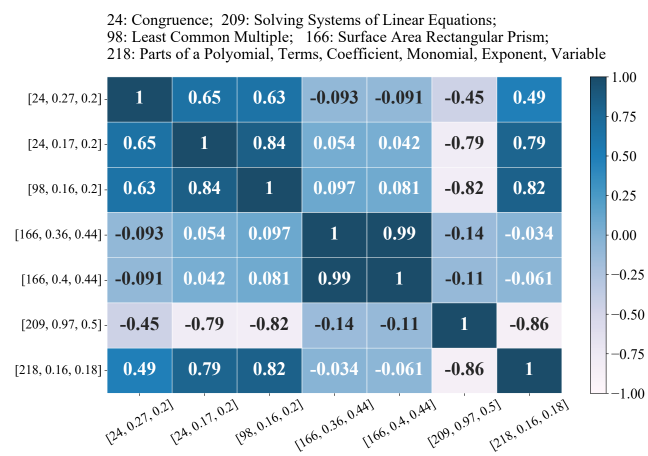


Figure 5: The relevant weights of seven different questions on the dataset ASSIST2012. Here the weights are the *Cosine Similarities* of the difficulty-enhanced question embedding learned by DIMKT. The label *[i, j, k]* means that a question has KC of *i* , question specific difficulty of *j,* and knowledge concept difficulty of *k.* For better understanding, we have normalized the range of difficulty value from 0 to 1, where a larger value means a harder question.

- DIMKT w/o ***KSU***, which refers to DIMKT without knowledge state updating module in ASNN.

We evaluate the above variant models on students' future performance prediction, and Figure 4 shows the experimental results, where we can find some inspired conclusions. Firstly, removing the subjective difficulty feeling in ASNN leads to the most significant performance drop on both datasets. Moreover, the other components (i.e., personalized knowledge acquisition and knowledge state updating) in ASNN are also necessary for maintaining the performance of DIMKT. These decreases are easy to understand, as these three components correspond to the critical modeling of the question difficulty effect on the three stages of the practice process. Secondly, we note that the personalized knowledge acquisition develops a more considerable impact on the dataset Eedi2020, i.e., it causes more performance declines in Eedi2020. The reason is that Eedi has a much larger number of students, so their personalized knowledge acquisition is more important for these students. Thirdly, both the *QS* difficulty and the *KC* difficulty contribute to the performance of DIMKT, while dropping the *QS* difficulty brings more declines as it is more capable of reflecting the question difficulty. Nevertheless, the *KC* difficulty is also essential auxiliary information to enhance the robustness of DIMKT. If we drop both of them, the performance of DIMKT will be worse.

5.7 Interpretability of the Difficulty-enhanced Question Embedding

In this section, we conduct some experiments to show the interpretability of the learned difficulty-enhanced question embedding in DIMKT. Specifically, we randomly chose seven questions with different KCs and difficulty levels on the dataset ASSIST2012. We then visualize the relevant weights between these questions by calculating their *Cosine Similarities* in Figure 5, from which we can see that the learned question embeddings reflect reasonable connections after adequate training. Figure 5 indicates that the relevant weights tend to be higher between questions related to the same concept or with closer difficulty levels. For example, the fourth question with a (KC, *QS* difficulty, *KC* difficulty) tuple of [166, 0.36, 0.44] and the fifth question of [166, 0.4, 0.44] are firmly related, and they have almost the same relevant weights to other questions, because they have the same KC (i.e., *166: Surface Area Rectangular Prism*) and only slightly different in the *QS* difficulty. Besides, although the third question of [98, 0.16, 0.2] and the last question of [218, 0.16, 0.18] have totally different KCs , there is a high weight between them as they have the same *QS* difficulty (i.e., 0.16). It is worth noting that the sixth question of [209, 0.97, 0.5] has significantly low relevant weights to all other questions, as its difficulty level is evidently higher. More similar cases can be easily found, such as the first question of [24, 0.0.27, 0.2] and the second question of [24, 0.17, 0.2], the second question of [24, 0.0.17, 0.2] and the third question of [98, 0.16, 0.2].

5.8 Visualization of Students' Performance Prediction Process

In Figure 6, we depict the performance prediction process of DIMKT on two students in the dataset Eedi2020. From Figure 6, we can see that DIMKT shows superior interpretability, i.e., it gives not only accurate predictions but also the reason that why students will succeed or not. For example, although student s_1 got a wrong answer on the questions with KC of *55: Transformations* and *QS* difficulty of 0.56 and 0.43 at the beginning, DIMKT successfully predicts that s_1 can correctly answer the question with the same

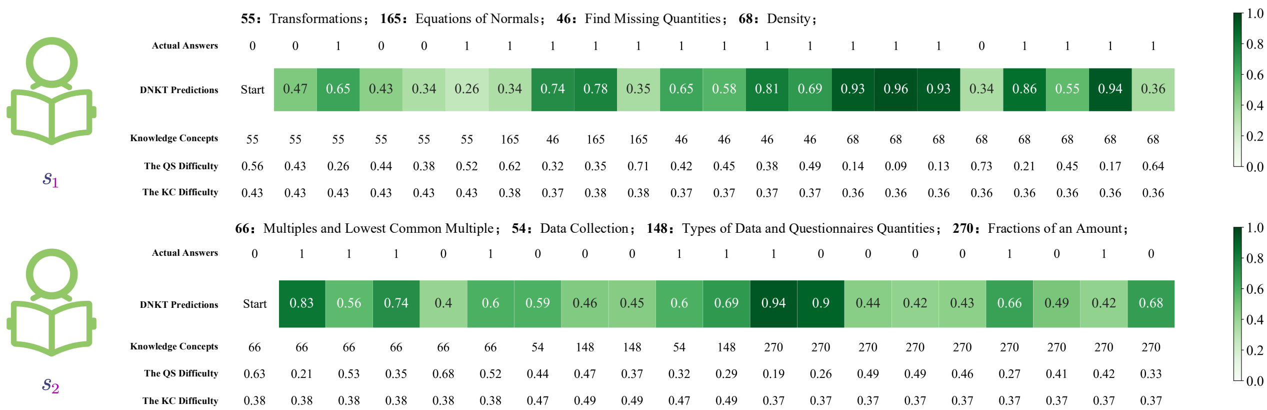


Figure 6: The learning sequences and performance predictions of two students on different KCs with various difficulty levels on the dataset Eedi2020. We have normalized the range of the QS and KC difficulty from 0 to 1, where the larger value means a harder difficulty.

KC of *55: Transformations* and QS difficulty of 0.26 because the latter question is easier. Besides, even s_1 has got right answers on questions with KC of *68: Density* and QS difficulty of 0.14, 0.09, and 0.13, DIMKT successfully predicts that s_1 would not pass the question with the same KC but harder QS difficulty of 0.73. Similarly, for student s_2, the predictions of DIMKT indicate the same interpretability. However, we note that his/her performance on questions with KC of *270: Fractions of an Amount* fluctuated greatly, and the DIMKT's predictions were not consistent with the actual answers many times. We guess there is a good chance of abnormal learning behavior for s_2, such as guessing and slipping. In general, DIMKT's predictions have well interpretability, which are in line with our cognition, i.e., *the lower the difficulty, the greater the probability of correct answers.*

6 CONCLUSIONS AND FUTURE WORKS

In this paper, we first analyzed the significant impacts of the question difficulty effect on learning. Then, we presented a novel DIfficulty Matching Knowledge Tracing (DIMKT) model to measure the question difficulty effect and improve KT performance by establishing the relationship between student knowledge state and question difficulty level. Specifically, we first directly utilized the question specific difficulty and the knowledge concept difficulty to enhance the question embedding. Then, we designed an Adaptive Sequential Neural Network (ASNN) to capture the connection between knowledge state and question difficulty level in DNKT. Specifically, by comparing the differences of the difficulty-enhanced question embedding and the student's knowledge state, ASNN first obtained students' subjective difficulty feelings before practice. Then, ASNN used the subjective difficulty feeling and the answer to calculate the student's knowledge acquisition during practice. After the practice, ASNN further developed a knowledge indicator to determine the student's updated knowledge state. Finally, we conducted extensive experiments to validate the effectiveness of DIMKT, and the results demonstrate that DIMKT achieved the new state-of-the-art performance. Moreover, DIMKT predicted not only what students' answers will be but also why they succeed or not, which is more instructive for promoting learning. In the future, we will try to utilize the question difficulty in more scenarios. For example, considering the question difficulty, we can infer students' slipping and guessing behaviors in learning.

ACKNOWLEDGMENT

This research was partially supported by grants from the National Key Research and Development Program of China (Grant No. 2021YFF0901003), the National Natural Science Foundation of China (Grants No. U20A20229, No. 61922073, and No. 62106244).

REFERENCES

[1] Sivakumar Alagumalai and David D. Curtis. 2005. *Classical Test Theory.* Springer Netherlands, Dordrecht, 1–14.
[2] 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.* ACM Press.
[3] Joseph Beck, Mia Stern, and Beverly Park Woolf. 1997. Using the student model to control problem difficulty. In *User Modeling.* Springer, 277–288.
[4] 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.
[5] Albert T Corbett and John R Anderson. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI* 4, 4 (1994), 253–278.
[6] J. Cowan. 2013. A learner centered approach to online education. *British Journal of Educational Technology* 44, 6 (2013), E221–E222.
[7] Kobe Desender, Filip Van Opstal, and Eva Van den Bussche. 2017. Subjective experience of difficulty depends on multiple cues. *Scientific reports* 7, 1 (2017), 1–14.
[8] Michel C Desmarais and Ryan S Baker. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-adapted Interaction* 22, 1 (2012), 9–38.
[9] Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *USER-ADAP* 19, 3 (2009), 243–266.
[10] Aritra Ghosh, Neil Heffernan, and Andrew S. Lan. 2020. Context-Aware Attentive Knowledge Tracing *(KDD '20).* New York, NY, USA, 2330–2339.
[11] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS.* 249–256.
[12] Le An Ha, Victoria Yaneva, Peter Baldwin, and Janet Mee. 2019. Predicting the Difficulty of Multiple Choice Questions in a High-stakes Medical Exam. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications.* Association for Computational Linguistics, Florence, Italy, 11–20.

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

[14] Zhenya Huang, Qi Liu, Enhong Chen, Hongke Zhao, Mingyong Gao, Si Wei, Yu Su, and Guoping Hu. 2017. Question Difficulty Prediction for READING Problems in Standard Tests. In *AAAI*.

[15] Gwo-Jen Hwang. 2003. A conceptual map model for developing intelligent tutoring systems. *Computers & Education* 40, 3 (apr 2003), 217–235.

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

[17] Barbel Knäuper, Robert F Belli, Daniel H Hill, and A Regula Herzog. 1997. Question difficulty and respondents' cognitive ability: The effect on data quality. *JOURNAL OF OFFICIAL STATISTICS-STOCKHOLM-* 13 (1997), 181–199.

[18] Feifei Kou, Junping Du, Yijiang He, and Lingfei Ye. 2016. Social network search based on semantic analysis and learning. *CAAI Transactions on Intelligence Technology* 1, 4 (2016), 293–302.

[19] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[20] Qi Liu, Shuanghong Shen, Zhenya Huang, Enhong Chen, and Yonghe Zheng. 2021. A Survey of Knowledge Tracing. *arXiv preprint arXiv:2105.15106* (2021).

[21] Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. 2019. Exploiting Cognitive Structure for Adaptive Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 627–635.

[22] Yunfei Liu, Yang Yang, Xianyu Chen, Jian Shen, Haifeng Zhang, and Yong Yu. 2020. Improving Knowledge Tracing via Pre-training Question Embeddings, Christian Bessiere (Ed.). International Joint Conferences on Artificial Intelligence Organization, 1577–1583.

[23] Derek Lomas, Kishan Patel, Jodi L. Forlizzi, and Kenneth R. Koedinger. 2013. Optimizing challenge in an educational game using large-scale design experiments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM.

[24] 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.

[25] Frederique M Lord. 1980. *Applications of Item Response Theory to Practical Testing Problems.* LAWRENCE ERLBAUM ASSCCIAATES.

[26] Frederic M Lord. 2012. *Applications of item response theory to practical testing problems.* Routledge.

[27] Ye Mao. 2019. One minute is enough: Early prediction of student success and event-level difficulty during novice programming tasks. In *In: Proceedings of the 12th International Conference on Educational Data Mining (EDM 2019)*.

[28] Sein Minn, Feida Zhu, and Michel C Desmarais. 2018. Improving knowledge tracing model by integrating problem difficulty. In *2018 IEEE International conference on data mining workshops (ICDMW)*. IEEE, 1505–1506.

[29] Ulrike Pado. 2017. Question Difficulty – How to Estimate Without Norming, How to Use for Automated Grading. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

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

[31] Shalini Pandey and Jaideep Srivastava. 2020. RKT: Relation-Aware Self-Attention for Knowledge Tracing *(CIKM '20)*. New York, NY, USA, 1205–1214.

[32] Jan Papoušek, Vít Stanislav, and Radek Pelánek. 2016. Impact of question difficulty on engagement and learning. In *International Conference on Intelligent Tutoring Systems*. Springer, 267–272.

[33] Zachary A Pardos, Yoav Bergner, Daniel T Seaton, and David E Pritchard. 2013. Adapting Bayesian Knowledge Tracing to a Massive Open Online Course in edX. *EDM* 13 (2013), 137–144.

[34] Zachary A. Pardos and Neil T. Heffernan. 2011. KT-IDEM: Introducing Item Difficulty to the Knowledge Tracing Model. In *User Modeling, Adaption and Personalization*. Vol. 6787. Berlin, Heidelberg, 243–254.

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

[36] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. In *NeurIPS*. 505–513.

[37] Liu Qi, Huang Zhenya, Yin Yu, Chen Enhong, Xiong Hui, Su Yu, and Hu Guoping. 2021. EKT: Exercise-Aware Knowledge Tracing for Student Performance Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[38] 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.

[39] 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. *SIGIR '20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval Virtual Event China July, 2020* (2020), 1857–1860.

[40] 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.

[41] Shiwei Tong, Qi Liu, Wei Huang, Zhenya Huang, Enhong Chen, Chuanren Liu, Haiping Ma, and Shijin Wang. 2020. Structure-based Knowledge Tracing: An Influence Propagation View. In *Proceedings of the The 19th IEEE International Conference on Data Mining*.

[42] Wim J Van Der Linden and Ronald K Hambleton. 1997. Item response theory: Brief history, common models, and extensions. In *Handbook of modern item response theory*. Springer, 1–28.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

[44] 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.

[45] Chenyang Wang, Weizhi Ma, Min Zhang, Chuancheng Lv, Fengyuan Wan, Huijie Lin, Taoran Tang, Yiqun Liu, and Shaoping Ma. 2021. Temporal Cross-Effects in Knowledge Tracing *(The International Conference on Web Search and Data Mining)*. Association for Computing Machinery, New York, NY, USA, 517–525.

[46] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. 2020. Neural Cognitive Diagnosis for Intelligent Education Systems. In *AAAI 2020*.

[47] Zichao Wang, Angus Lamb, Evgeny Saveliev, Pashmina Cameron, Yordan Zaykov, José Miguel Hernández-Lobato, Richard E Turner, Richard G Baraniuk, Craig Barton, Simon Peyton Jones, Simon Woodhead, and Cheng Zhang. 2020. Diagnostic questions: The neurips 2020 education challenge. *arXiv preprint arXiv:2007.12061* (2020).

[48] Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation* 1, 2 (1989), 270–280.

[49] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (jan 2021), 4–24.

[50] Chun-Kit Yeung and Dit-Yan Yeung. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*. ACM.

[51] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *WWW*. 765–774.

[52] Moyu Zhang, Xinning Zhu, Chunhong Zhang, Yang Ji, Feng Pan, and Changchuan Yin. 2021. Multi-Factors Aware Dual-Attentional Knowledge Tracing. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. Association for Computing Machinery, New York, NY, USA, 2588–2597.