Dynamic Learner's Knowledge Assessment by Incorporating Learner and Domain Modeling in Intelligent Tutoring Systems

by

Wenbin Gan

Dissertation

submitted to the Department of Informatics

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

SOKENDAI

The Graduate University for Advanced Studies, SOKENDAI

March 2022

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Yuan Sun, for giving me the opportunity to carry out my research in Japan as a Ph.D. student. It is always her continues guidance, encouragement and support that help me overcome all the difficulties and bottlenecks in my research and my study. She introduced some researchers in the field to me, provided me the opportunities to discuss with them and find the key research problems in this thesis. She also gives me inexhaustible patience to help me process all kinds of matters in my daily life, like a friend and an elder in my family, which makes my life in this foreign country warmly and invaluable. I really enjoy and value these three years here.

I would like to thank Prof. Masaki Nishizawa and Prof. Akiko Aizawa for providing me the research equipment that greatly facilitates my research. They also given me a lot of suggestions and ideas on my research. I would like to thank Prof. Yi Yu and Prof. Ryutaro Ichise for the valuable suggestions on my work and the presentations. I would like to thank Prof. Hiroaki Ogata from Kyoto University, who is a member of my dissertation committee, for all his technical comments on this thesis. I would like to thank Prof. Yi Sun, from University of Chinese Academy of Sciences, who gave me constructive suggestions on the work in these years. We have regular group meeting, and this facilitates my understanding of the research trends in this field.

I would like to thank all colleagues and friends in NII, as well as all the friends in Tokyo International Exchange Center (TIEC). There is so much happy time with all of you. Thank you all for being in my life.

I also express my gratitude to SOKENDAI and NII for providing the best research environment. Also thank the International Affairs and Education Support Team for their consistent help.

I would like to thank Japan Society for the Promotion of Science (JSPS) for supporting me as a JSPS research fellow. I couldn't complete the Ph.D. course without the support from JSPS.

Lastly, I must thank my families for their love, support, and unshakable trust in me. They gave me the confidence to trust in myself and work hard to attain my dreams. Their continual supports provided me the motivation to fight through the COVID-19 pandemic and complete this dissertation. I couldn't have come this far without their support.

This work is supported by JSPS KAKENHI with Grant Number 18K11597 and the Grant-in-Aid for JSPS Research Fellows 20J15339.

Abstract

The popularity of online learning has increased in recent years, with increasingly many intelligent tutoring systems (ITSs) becoming available to learners. In general, these platforms enable learners to acquire knowledge in the process of a series of individualized learning activities (e.g., exercising solving) that accommodate learners with different needs and knowledge proficiencies. A key technique underlying these adaptive tutoring services is learner knowledge assessment, which aims to model learner performance to discover their latent knowledge proficiency in mastering knowledge concepts in a domain.

This task can facilitate the optimization of human learning because the assessment information is fundamental for the further adaptive services in many real-world ITSs. For example, adaptive remedial learning materials can be automatically provided based on students' individual needs, and content that is predicted to be not in conformity with students' knowledge states can be skipped or delayed, thereby effectively improving their learning efficiency and avoiding any decrements in their engagement. Meanwhile, given the popularity of a growing number of online educational platforms, a large number of learning logs can be collected for the purpose of building advanced models for accurate learner knowledge assessment. This has been a popular interdisciplinary research topic across education, psychology, computer science, and cognitive science.

Nevertheless, discovering learners' latent knowledge state from the learning logs in an ITS is a rather challenging task, as human knowledge construction is a dynamic procedure and their knowledge is constantly evolving since learners dynamically learn and forget over time. Moreover, the knowledge attainment can be affected by many factors from both the learners and the learning domains. The existing studies have explored this task and proposed effective approaches from two directions: cognitive

diagnostic assessment (CDA) and knowledge tracing (KT). However, there are still numerous methodological issues, which restrict their practical applications. In this work, we address three important ones, namely, insufficient learning factor modeling, data sparseness and information loss, and fine-grained assessment and interpretability.

To tackle these issues, we proposed a general framework for dynamic learner knowledge assessment by integrating both learner and domain modeling. Based on this framework, we proposed three approaches, each addressing one specific issue in the existing studies. Specifically, on the first issue, we investigated the learner factors (learning and forgetting) and domain factor (item difficulty) by making use of rich information during learners' learning interactions and proposed a novel model named KTM-DLF that traces the evolution of learners' knowledge acquisition over time by explicitly modeling their learning and forgetting behaviors as well as the item difficulty. Extensive experiments confirmed the effectiveness of this model as it takes more and precise information into the modeling procedure. For the second issue, we explored to incorporate the knowledge structure (KS) into the learner assessment procedure to potentially resolve both the sparseness and information loss. We explored to automatically generate the KS from the learning logs and proposed a novel KS-enhanced graph representation learning model for KT with an attention mechanism (KSGKT). Extensive experiments demonstrated the superiority of the KSGKT model and the results proved it to be a good trial to alleviate the data sparseness and the information loss in conducting learner knowledge assessment. To cope with the third issue, we proposed a dynamic CDA model called KIEDCDA that incorporates not only the ability to trace the evolution of learners' knowledge proficiencies over time for large-scale assessments, but also the interpretability to explain learner performance in terms of their current knowledge proficiency and item characteristics. Experiments on several real-world datasets demonstrated the superiority and interpretability of the KIEDCDA model for learner performance modeling, suggesting that it is worthy of a good trial to track and explain learners' fine-grained and evolving knowledge states simultaneously.

This research has several contributions to the entire ITS field. It explored the task of dynamic learner knowledge assessment to obtain the individual learner's evolving knowledge states, which is the pillar of learner characteristics in ITS. The distribution of a learner's knowledge states provides a distinctive latent profile of the learner for the

ITSs, and lets the ITSs know who they are teaching, hence increasing the adaptability and individualization of the further services. Moreover, this thesis modeled various aspects of the learning domain, the obtained characteristics of learning content are essential for the ITSs to manage learning content and help the ITSs understand what they are teaching. Furthermore, this thesis investigated the potential of educational data mining driven decision-making in ITSs for adaptive online tutoring, and explored to track and explain learners' evolving knowledge states simultaneously. This will be helpful for the further tutoring services and provides ideas for explainable feedback.

This thesis presents our trials on dynamic learner knowledge assessment that enhanced the existing techniques. To further stimulate new ideas in the field of intelligent education, some remaining issues and future work are also described.

Contents

List of Figures

List of Tables

1 Introduction

Amid the COVID-19 pandemic, online education has become an inevitable choice and a mainstay in many nations [6, 7]. Recently, online learning has been more integrated with artificial intelligence approaches as a result of fast technological advancements, allowing for the development of more individualized educational systems. These systems are known as Intelligent Tutoring Systems (ITSs) [8]. With the explosive growing number of learners enrolled in online learning, the big data accumulated by these ITSs provides the potential to analyze and infer the latent factor/characteristics of the online learners and the learning resources, which is essential and fundamental for the systems to provide adaptive services [9, 10, 11, 12, 13].

This thesis is about the techniques that make the ITSs more intelligent and equip them with adaptive capabilities from the perspective of learning analytics on the big educational data. Specifically, this thesis is dedicated to the methods that dynamically assess the knowledge of online learners during their interaction with the systems, a fundamental personalized-tutoring technique underlying the adaptive services in ITSs.

This chapter briefly explains the background of this thesis in Section 1.1. It then gives an overview of this thesis in Section 1.2, explaining the motivation and the topic, the potential issues with existing methods and models, and the contributions. The thesis outline is described in Section 1.3.

1.1 Background: ITSs

Online learning has become increasingly popular in recent years, with proliferated online learning platforms becoming available to learners [14, 15, 16]. In general, these platforms enable the knowledge-acquisition of learners through a series of learning activities (e.g., video lecturing and exercising solving), to improve their formal and informal learning experiences.

There are two general forms of online learning [17]: one is based on the ITSs that involve the (semi-)autonomous computer programs, such as Carnegie Learning[1], ASSISTments[2], Knewton[3], Riiid TUTOR[4] and ALEKS[5], to provide tutoring feedback to learners; The other is the person-to-person online tutoring that enables human tutors to provide instructions to learners via communication tools, such as Zoom[6] and Webex[7], in a synchronous or asynchronous way. This thesis is concerned with the first form of online learning through ITSs, with the goal to make the ITSs more intelligent and equip them with adaptive capabilities.

ITSs, “computer programs that are designed to incorporate techniques from the AI community in order to provide tutors which know what they teach, who they teach and how to teach it” [18, 19], are a critical category of a carrier of online learning [6] which refers to using artificial intelligence techniques to facilitate online learning through structuring numerous resources and assisting learners in acquiring knowledge online, etc. These systems mimic individualized human tutoring in a computer-based environment [20], and are capable of offering delicate instructions during problem solving, tracking learners’ abilities and knowledge acquisition, and recommending

[1] http://www.carnegielearning.com/
[2] https://new.assistments.org/
[3] https://www.knewton.com/
[4] https://www.riiid.co/toeic-online-test/
[5] https://www.aleks.com/
[6] https://zoom.us/
[7] https://www.webex.com/

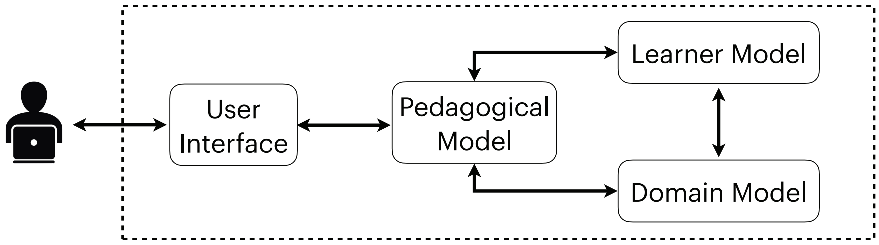


Figure 1.1: General ITS architecture.

learning resources to accommodate for the different learning preference and aptitude of individual learners [21]. ITSs have a particular strength in dealing with the interactive and personalized aspects of individual learning effectively [20], thus providing an alternative to the "one-size-fits-all" approach in traditional web-based learning [21].

The general architecture for the ITSs is plotted in Figure 1.1. It mainly consists of three models and a user interface [18, 22]. The user interface presents the learning interactions between learners and the ITSs in different forms, such as the learning materials, hints, feedback from the system, etc. The three models in the ITSs are described as follows:

- **Learner model**: it is the core component of an ITS. It provides an ITS with the information of who it teaches, which is the fundamental for the adaptivity of the ITS. To provide the personalized and precise tutoring services to a specific learner, an ITS should build a learner profile that contains as much explicit and implicit knowledge as possible about the learner's cognitive and affective states and their dynamic evolution during the long-term learning process [22]. The cognitive and affective states built in a learner profile are individualized factors such as the level of knowledge, activities, responses, behaviors, learning styles, preferences, cognitive engagement, mood and emotion and other information about a learner inferred and updated from the accumulated logs during the interaction process with the system [8].
- **Domain model**: it provides an ITS with the information of what it teaches. It contains the expert knowledge (concepts, rules, and problem-solving strategies, etc.) to be learned in a specific domain. This domain knowledge is generally organized into curriculums with some structures (hierarchies, networks, frames, etc.) that link the knowledge together according to pedagogical sequences [22].

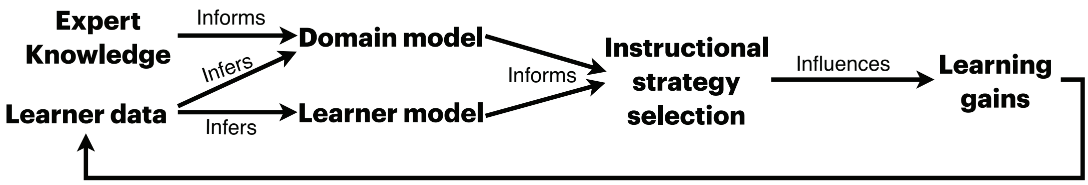


Figure 1.2: The data-driven ITS for personalized tutoring.

- **Pedagogical model**: it provides an ITS with the strategies of how to teach. Taking the input from both the domain and learner models, it makes decisions about tutoring strategies and actions to be taken [22].

The existing ITSs usually contain part or all of these models in the back-end with different levels of intelligence. Recently, ITSs have become increasingly integrated with artificial intelligence techniques in order to provide more personalized tutoring. With the explosive growing number of learners using the ITSs, the big data accumulated by these systems provides the potential to build more accurate learner models, to automatically structure and label the expert knowledge in the domain, and to learn the precise pedagogical models from previous experiences in a data-driven matter, as shown in Figure 1.2, which is essential and fundamental for the ITSs to provide adaptive and personalized services to individual learners.

1.2 Thesis Overview

1.2.1 Motivation

This thesis focuses on the techniques that automatically build the learner and domain models in the ITSs from the perspective of learning analytics on the big educational data, assuming the pedagogical model is available. More specifically, this thesis deals with the issue of *dynamic learner knowledge assessment (DLKA)* to obtain each individual learner's evolving knowledge states, which indicate their mastery of the particular knowledge in a domain. This is the most typical type of learner modeling as well as being the fundamental model type in ITSs [23].

Figure 1.3 shows an instantiated ITS architecture of the general architecture in Figure 1.1. Three modules (learner assessment module, problem domain module and

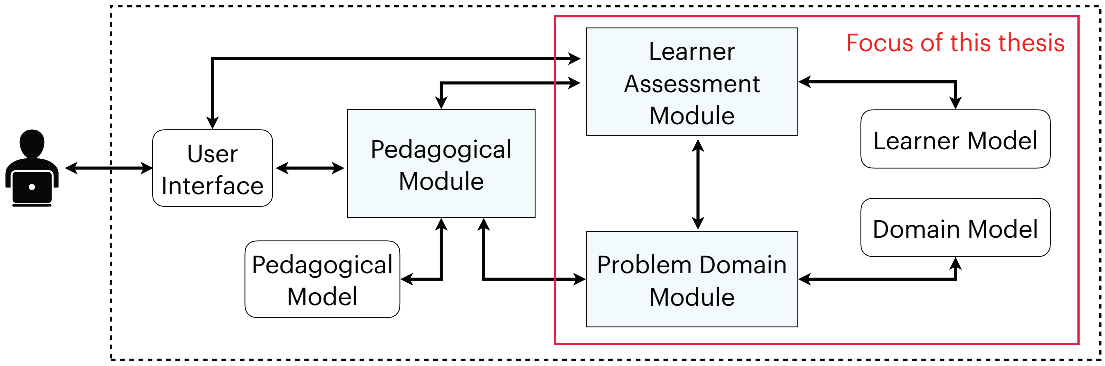


Figure 1.3: The focus of this thesis under a more specific ITS architecture.

pedagogical module) derived from the three models are added and linked in the figure and they form a more specific ITS architecture. In fact, learner knowledge assessment is only a part of learner modelling, researcher have conducted various researches on learner modeling under different sub-topics, such as learner cognitive engagement [24], learning style [25], learning emotion [26], etc.

As mentioned above, the aim of this thesis is to dynamically assess the knowledge states of learners, which is a typical type of learner model. Generally speaking, domain model and learner model are viewed to be separate; however, in this thesis we try to propose approaches for modeling both where the learning logs of learners can be useful indications for unfolding the latent structure of a learning domain [27]. Moreover, the process of learner knowledge assessment is based on the interaction between learners and the learning materials, hence the knowledge assessment is inextricably related to the domain modeling (e.g., the definition of knowledge components, the mapping of the learning materials to these knowledge components, and the modeling of content difficulty) [23]. Therefore, this thesis conduct DLKA by incorporating both learner and domain modeling, as shown in the red box of Figure 1.3.

Given the learners' long-term interaction logs in an ITS, the DLKA task is generally formulated as the problem of modeling their performances and inferring the evolving knowledge states that they accumulated from the interaction process in a long time. DLKA is essential for personalized learning and also plays a fundamental role in ITS [28, 9, 29, 16, 30]. The results of DLKA can optimize human learning in many real-world learning systems because the assessment information is fundamental for many further adaptive services [27, 31, 32, 29, 33]. Based on the inferred knowledge states of learners, tailored learning activities and support can be provided to meet individual learning needs and fulfil the diverse capabilities of learners. Content that is predicted to be not in conformity with learners' knowledge level can be skipped or delayed, thereby effectively improving their learning efficiency [23, 34] and avoiding any decrements in their engagement [35]. Moreover, a timely intervention of learning procedures by designing new measures and learning materials to remedy the weakness of learners can help teachers and administrators. Learner themselves can also better understand their weakness and strength and pay more attention to the poorly mastered knowledge concepts for better self-regulated learning. Meanwhile, given the popularity of a growing number of online ITSs, a large number of learning logs can be collected for the purpose of building advanced models for accurate DLKA [36, 37]. This has been a popular interdisciplinary research topic across education, psychology, computer science, and cognitive science [16, 38].

Nevertheless, discovering learners' latent knowledge state from the long-term learning logs in an ITS is a rather challenging task. From the perspective of human learning, some of the major concerns are listed below:

- Dynamic knowledge construction procedure. Human knowledge construction is a dynamic procedure and is constantly evolving as learners dynamically *learn* and *forget* over time. ITSs provide learners with abundant learning resources and enable them to learn individually at their own pace. knowledge acquisition is realized through this continuous learning process, and inevitably, learners will forget the knowledge they have learned. The knowledge construction procedure is a trade-off between acquiring new knowledge and forgetting old one; the knowledge retention is fluctuated continuously with space scheduling in human learning.
- Complex human knowledge attainment. The knowledge attainment process can be affected by many factors (affect, motivation, identity, etc.) at both the macro and micro level [11]. Moreover, learning can occur during explicit contexts, such as classroom teaching and exercising solving, or can occur implicitly by experience in our external world [39], not to mention the epiphany in human learning. In addition, there are complex relationships between the learning activities and the knowledge acquiring outcomes.

- Latent and non-explicit knowledge state. It is obviously that learners' knowledge states are latent and cannot be directly observed and quantified due to the complexity of human brain, which leads to the inaccessibility of the ground-truth.

From the perspective of implementing DLKA approaches, the main challenges are as follows:

- Dynamic learning process modeling. As we mentioned above, human learning process is dynamic and involves complex knowledge construction procedure as learning and forgetting occur at the same time. How to build models to adequately model the cognitive process of this complexity and dynamics in a longitudinal manner is an extremely challenging issue. Moreover, learners' performance in the future is deeply influenced by their long-term historical learning experiences, especially on their learning of the related knowledge states. Capturing these long-term dependencies in the learning sequences [31] to precisely monitor the evolution of learners' knowledge states is another big challenge.
- Fine-defined domain modeling. Learning domain is inextricably related with the development of learner knowledge. The learning procedure involves the acquaintance and application of the massive knowledge, and the knowledge in a domain is generally decomposed into a set of small-granularity knowledge components [40]. In the practical learning scenarios, these knowledge components are embedded in the individual learning objects, such as exercises and learning videos, to practice learners' mastery on specific knowledge components. A realistic ITS usually contains hundreds of knowledge components and tens of thousands of learning objects. The knowledge components and the learning objects have complex relations between each other, e.g., the prerequisite relations and similarity. How to organize and characterize the knowledge components and the large numbers of learning objects in a domain is essential for the assessment of learner knowledge.
- Sparseness and noise of the learning data. In an ITS, a learning object only incorporates one or several knowledge components, and the number of learning objects is far greater than the number of knowledge components, most learners only attempt a small part of the learning objects with higher dropout rate. Accordingly, the response data are quite sparse [41, 42]. Moreover, some learners may game the systems with a set of non-learning-oriented strategies [43], which results in noise in the collected

learning data.

- Interpretability. The interpretability is an important factor that accounts for good tutoring services for self-regulated learning. Learners wonder not only what they need to learn further but also why they need to learn it, i.e., acquiring the degree of mastery on some certain knowledge concepts. However, it is nontrivial to both quantify the impacts of long-term content learning on improving the knowledge acquisition and enable the interpretability of the DLKA models [1].

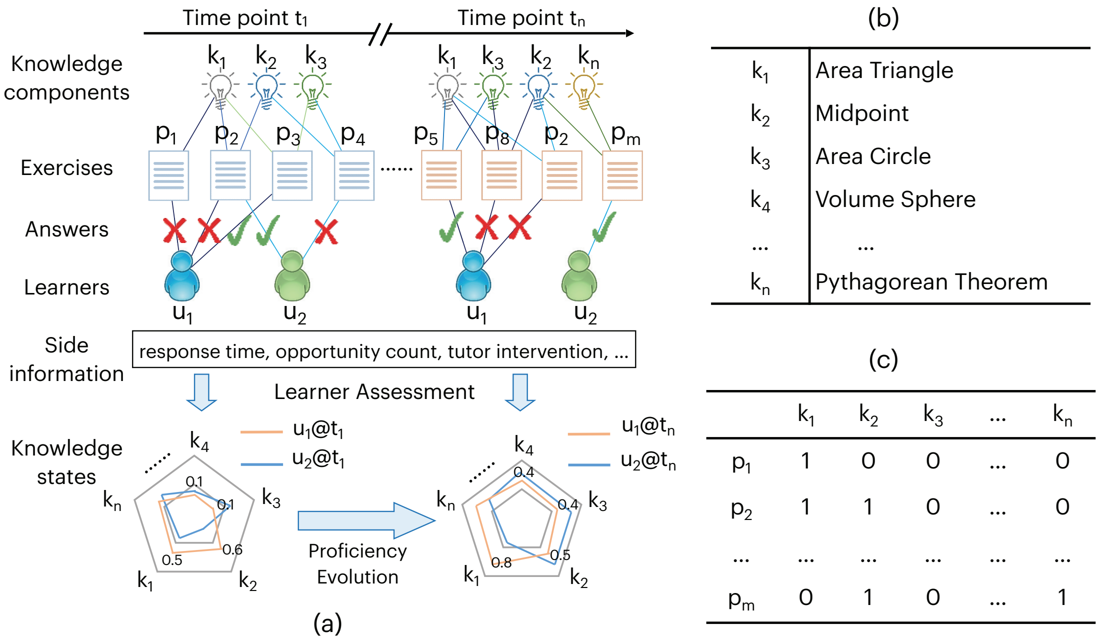
This thesis is dedicated to propose novel DLKA models by considering these challenges both in human learning and also in the practical implementation on the long-term learning data. In the next section, we show the research topic and scope.

1.2.2 Topic and Scope

The problem of DLKA primarily focuses on monitoring learners' evolving degrees of mastery on various conceptual or procedural knowledge components (KCs)[8] in a specific domain [9, 10]. However, as mentioned above, learners' knowledge states on KCs are latent and cannot be directly observed and quantified. To cope with this issue, we need to turn to some observable indications to infer their knowledge states.

Researchers in this field have worked on this topic by utilizing learners' explicit learning feedback on various types of learning materials, such as readings [44], video lectures [45], assignments, exercises [9, 10, 11] and discussions, as well as the multiple learning resource types concurrently [46, 47, 48]. Some types of learning materials are gradable, such as exercises and quizzes. A learner's grade can be interpreted as an explicit indication of learner knowledge when they engage with such kinds of materials. If a learner obtains a high mark on an item, it is probable that he or she has acquired the necessary KCs to solve that item. Others are not gradable and their influence on learner knowledge is indirect and implicit [47]. To obtain more explicit indication of learner knowledge, in this thesis we use the learners' explicit feedback on gradable *exercises* to assess learner knowledge, i.e., monitoring learners' evolving knowledge states based on the long-term exercising logs, which is the most common manner for implementing the DLKA models.

[8] KCs are atomistic components of knowledge in a domain; in cognitive psychology, KC is also termed as attribute or skill. In this thesis KC and skill will be used interchangeably.



(b)

k_1	Area Triangle
k_2	Midpoint
k_3	Area Circle
k_4	Volume Sphere
...	...
k_n	Pythagorean Theorem

(c)

	k_1	k_2	k_3	...	k_n
p_1	1	0	0	...	0
p_2	1	1	0	...	0
...	...	...	...	...	...
p_m	0	1	0	...	1

Figure 1.4: Showcase of the learner knowledge assessment task for the two users practicing the exercises in an ITS. (a) Assessment information on each learner's knowledge states on all knowledge concepts at each time point can be obtained using DLKA models; (b) the example KCs involved in these exercises; (c) the Q-matrix for mapping the specific exercises to these KCs.

Figure 1.4(a) shows a toy example for the DLKA task. Two learners [9](u_1 and u_2) attempt exercises [10] in a tutoring system in a certain time period. The KCs they want to master are shown in Figure 1.4(b). These KCs are embedded in the explicit exercises. In general, each exercise involves one or several KCs that are required to solve the exercise (e.g., "12-8=?" involves the KC of "subtraction of two integers" and "3.8-1.6=?" is related to "subtraction of two decimals"). This information is typically encoded in the form of a Q-matrix (as shown in Figure 1.4(c)) given as prior knowledge from education experts denoting which skills are required for each problem [33]. The number one in the Q-matrix indicates that the specific exercise involves the corresponding KC, and zero indicates otherwise. Given learners' learning logs (answers to exercises and other side information) in a system, the DLKA task is to model their performances and infer the knowledge states that they accumulate from the exercising process.

[9] "Learner" here is a more general concept, it is also referred as "student" and "user".

[10] We will interchangeably refer to exercises as questions, items or problems.

To more precisely formulate the problem, we now explain the topic of this thesis using a more technical wording:

> Let us suppose that a learning system has I students, J exercises, and K underlying knowledge components. Each exercise is associated with one or several KCs needed to solve the exercise, i.e., the system have provided the KC labelling in the Q-matrix. Each KC can be trained by attempting several exercises. Moreover, each learner can learn and perform exercises individually at different times, and will attempt an exercise correctly or incorrectly every time. Whenever a learner attempts an item at time point t, the system generates an exercise record $n_t = (u_i, q_t, s_k, r_t, si_t, ts_t)$, where n_t is a tuple including the learner ID $u_i \in \{u_1, u_2, ..., u_I\}$, exercise ID $q_t \in \{q_1, q_2, ..., q_J\}$, underlying KCs $s_k \in \{s_1, s_2, ..., s_K\}$, the correctness $r_t \in \{0, 1\}$ of the learner's answer at timestamp t, side information si_t during this interaction (e.g., the elapsed time spent on solving the given question and the opportunity count of attempting this question or KC), and the timestamp ts_t of the current time point t. A learner's exercising process is then modeled as a sequence $X = \{n_1, n_2, ..., n_T\}$, where T indicates the latest timestamp.

Note that we estimate learners' knowledge from their exercising logs, the smaller the gap between the estimated and learners' real knowledge states, the more accuracy of the built models [49], as shown in Figure 1.5. However, there is no ground-truth of the learners' real knowledge, the obtained assessment results cannot be directly evaluated. Researches from cognitive psychology, such as the Classical Test Theory and Item Response Theory [50], have verified that the effectiveness of the estimated learner knowledge can be validated by predicting learner scores on the exercises. Hence in the practical implementation, DLKA models are verified by comparing the predicted results and the real results of learners on the attempted exercises. Based on this, the existing researches on DLKA generally use the learner performance prediction to test their models, aiming at alleviating the gap between the estimated and the real learner mastery of domain knowledge.

Without loss of generality, the research topic can be formulated as follows:

(**Problem Formulation**) *Given a learner's past exercising sequence X in a system*

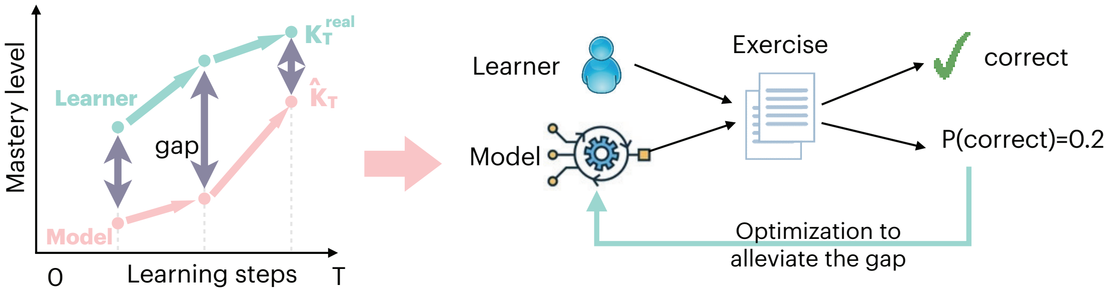


Figure 1.5: Learner knowledge assessment is evaluated by the results of learner performance prediction on the attempted exercises.

and a new exercise q_{t+1}, our goal is to model the learner–exercise interaction procedure, and hence track the evolution of a learner's knowledge states from timestamp 1 to T. Based on the estimated learner knowledge states, we then predict the probability $p(r_{t+1} = 1|X, q_{t+1})$ that the learner answers exercise q_{t+1} correctly.

To adequately capture the complexity of learners' cognitive process, this thesis is dedicated to propose rich DLKA model with non-linearity on the massive learning interaction data, with the emphasis on monitoring the dynamics of learner knowledge in a longitudinal manner. It does not assess the learners based on a certain quiz, such as GRE (Graduate Record Examinations) and TOEFL (Test of English as a Foreign Language), in a static way. Moreover, it is based on the big data of long-term learning logs, as will described in Chapter 3, it may be challenging for the proposed models to be directly applied to small data of classroom-level learners, which will be one of our future work.

This thesis solve the task of DLKA, assuming the definition of KCs, the designing of exercises in a domain and the mapping of specific exercises to these KCs (i.e., the q-matrix) are available. Actually, for many ITSs, this domain-dependent work is manually designed by domain experts, based on which the learning is conducted. Moreover, although the data collected for this thesis, as will described in Chapter 3, is mainly from the ITSs that are used for learning mathematics and English, the models proposed in this thesis are domain-independent, and can be directly applied to other domains.

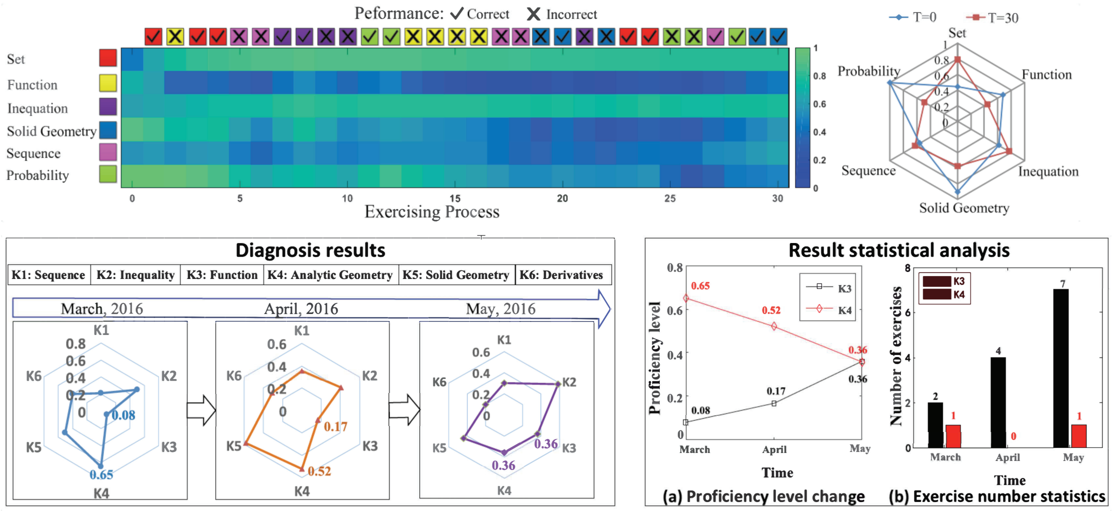


Figure 1.6: Fine-grained Diagnostic Report. The above figure shows a learner's changing knowledge states after attempting each exercise [1]; the bottom left figure shows the learner's knowledge states at fixed interval time from a macroscopic perspective [2]; while the bottom right figure shows the results of statistical analysis based on the learning logs, e.g., the attempt counts for specific KCs.

1.2.3 Example Application Scenarios

Fine-grained Diagnostic Report The research in this thesis can be directly applied to the tutoring systems by providing learners with fine-grained diagnostic report. Compared with the coarse-grained information such as correct/incorrect feedback or the score/rank of learners' exercise process, this fine-grained diagnostic report can be more helpful to learners when conducting the self-regulated learning [51, 33]. From a tutoring viewpoint, learners who understand the strengths and weaknesses of their knowledge points can remedy these weaknesses and improve themselves through self-regulated learning. From a teaching viewpoint, a comprehensive diagnostic report would help teachers identify the knowledge levels of both the whole class and individual students. Based on this information, they can design and provide timely interventions of the learning procedures.

Figure 1.6 shows an example diagnostic report provided by an ITS during the whole learning process. Assisted by this fine-grained diagnostic report, learners can focus on their weak knowledge without repeated training on their already mastered skills. This enlightenment will greatly improve students' learning efficiency.

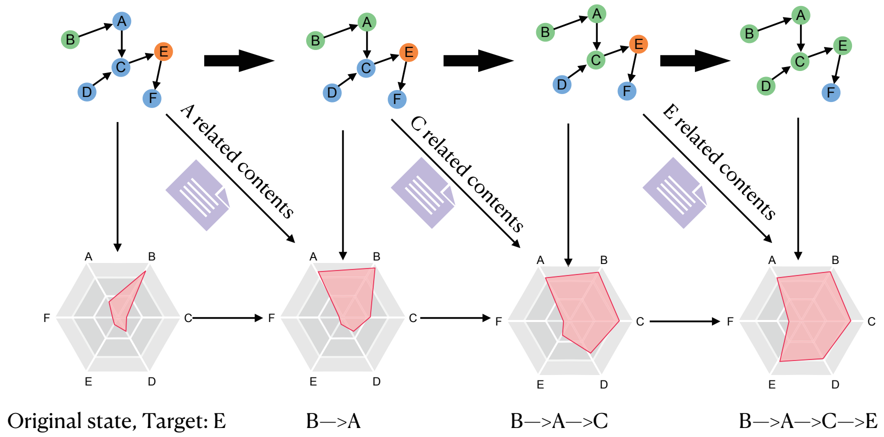


Figure 1.7: Adaptive learning-path recommendation based on the learners' knowledge level and the knowledge structure. The learning path "B→A→C→E" is recommended to a learner for mastering the target skill E . The green and orange nodes in the knowledge graph represent the already mastered skills and the target skills, respectively. The below-radar graphs show the changes in the learner's knowledge states on each skill. Gradually, the learner mastered all skills in the learning path.

Adaptive Learning-Path Recommendation Adaptive learning-path recommendation reasonably arranges the order of the learning contents to generate a well-defined learning path. Along this path, a learner can efficiently complete the learning target and alleviate the information overload issues in e-learning [52]. The research in this thesis can be directly applied to this task. Incorporating the inferred knowledge states of learners and the knowledge structure in the domain, the adaptive learning-path recommendation service can be provided. In the example of Figure 1.7, the learner is requested to master skill "E" along the recommended learning path "B→A→C→E" based on his or her current knowledge level. This path follows the logicality determined by the knowledge structure [53]. As the learner has already mastered skill "B", he or she approaches skill "C“ from skill "B" rather than from skill "D". Along the learning path, the learner is provided with the corresponding learning contents. After completing this self-learning process, the learner has gradually mastered all skills along the learning path. Such self-awareness can greatly improve the adaptive navigation ability of existing ITS.

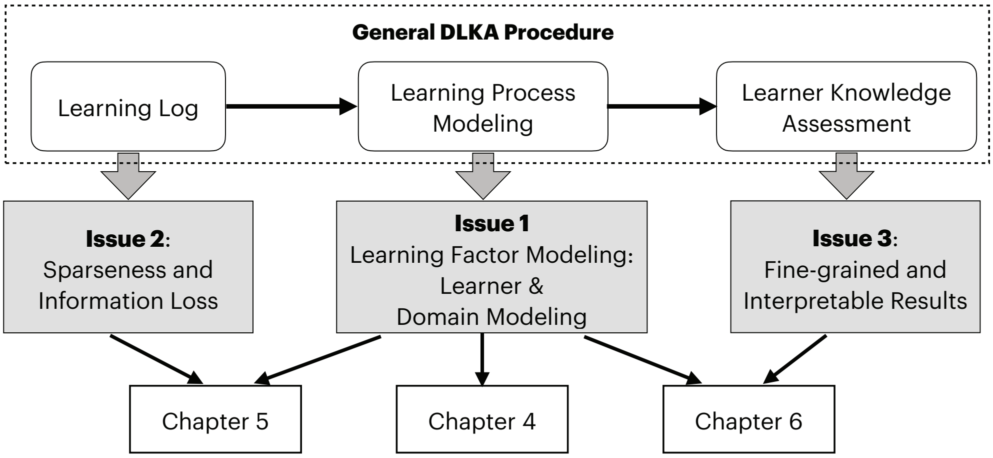

General DLKA Procedure
Learning Log
Learning Process Modeling
Learner Knowledge Assessment
Issue 2: Sparseness and Information Loss
Issue 1 Learning Factor Modeling: Learner & Domain Modeling
Issue 3: Fine-grained and Interpretable Results
Chapter 5
Chapter 4
Chapter 6

Figure 1.8: The three issues solved in this thesis.

1.2.4 Issues to Be Addressed

We have introduced the general procedure of DLKA in § 1.2.2. For better understanding, we show this general procedure in Figure 1.8. Given the learning log data, learning factors that influence the learning performance are quantified to capture the impact on knowledge acquisition during the learning process, and then learner knowledge assessment is conducted by integrating all these factors to monitor learners' evolving knowledge states over time. Researchers in this field have proposed various models to implement this procedure. Although the existing researches have achieved good performance on this task, some important issues remain (partially) unsolved. This thesis analyzes the existing learner knowledge assessment approaches and identifies three potential issues (as shown in the three shaded boxes in Figure 1.8):

1. Issue 1: what factors influence the learning performance and how to quantify these factors and utilize them to model the dynamic evolution of learner knowledge?

2. Issue 2: How to alleviate the data sparseness and the information loss in conducting learner knowledge assessment?

3. Issue 3: how to track and explain learners' fine-grained and evolving knowledge states simultaneously?

As we assess learners' evolving knowledge from their exercising logs, the first and most important consideration is to find the factors that result in the change of their knowledge acquisition, based on which the knowledge evolution process is modeled. Cognitive psychology has long verified that the knowledge acquisition procedure is not only related to the learners but also the learning materials in the domain. Existing approaches for DLKA consider only a fragment of the information during the learning process that results in the change of learner knowledge acquisition. To more precisely assess the learner knowledge, we must first pinpoint the factors that influence the evolving knowledge and propose methods to quantify them. This issue is explored in Chapter 4. With issue 1 addressed, we find that the performances of existing approaches greatly suffer from the sparseness of the input data and the information loss when modeling the learning process, hence we solve this issue (issue 2) in Chapter 5 of this thesis. We propose deep learning models in this thesis to model the learning process by leveraging the powerful representation ability of the deep neural networks in a data-driven manner, however, deep neural network is regarded as a black-box, how to track and explain learners' evolving knowledge states simultaneously is an important issue. Moreover, the interpretability is an important factor that accounts for good tutoring services in an ITS. This issue will be solved in Chapter 6.

1.2.5 Contributions

In this thesis, we propose a general framework, used as a general idea for solving the research task in § 1.2.2. This framework is then instantiated to three approaches, each addressing the issues in § 1.2.4 from different perspectives. In this section, we introduce the philosophy of this framework and then give an overview of the contributions.

As shown in Figure 1.9, the learners interact with the exercises in the tutoring systems, and their exercising results are recorded in the learning logs. Existing work generally performs DLKA on these exercising results to monitor the learners' evolving knowledge. However, the exercising procedures are also very important for the task of learner knowledge assessment. Cognitive psychologists have long verified that the procedure of knowledge acquisition is linked with multiple factors that not only related with the learners (e.g., learning and forgetting) but also the learning materials

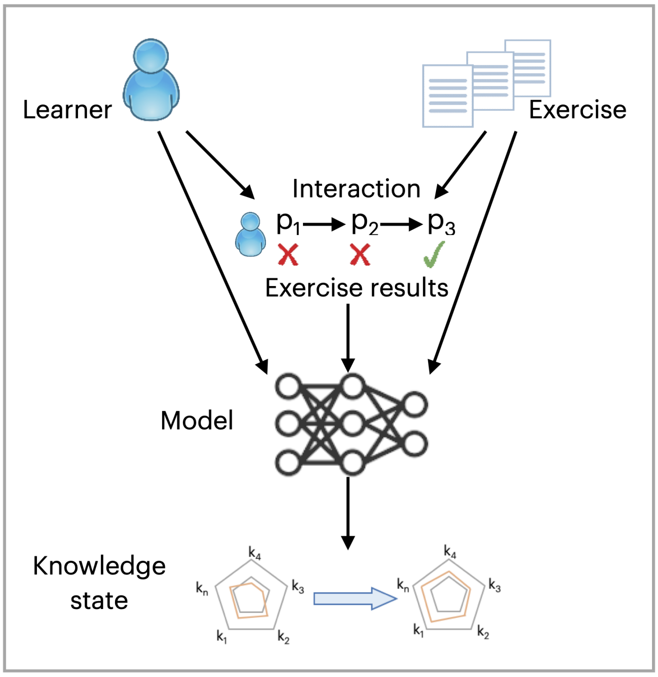


Figure 1.9: The general framework for DLKA.

(e.g., difficulty and discrimination) [54, 55, 56]. Hence in this thesis, we solve the DLKA task by incorporating both learner and domain modeling.

This framework is then instantiated to three approaches that address the above-mentioned issues from different perspectives. Specifically, the contributions of this thesis are as follows:

- A deep factorization machine based approach for learner knowledge assessment by modeling multiple factors. It solves the Issue 1 by exploring the factors that influence the knowledge acquisition and making use of rich information during learners' learning interactions to achieve more precise prediction of learner knowledge. In Chapter 4, we propose a novel knowledge tracing model named KTM-DLF that traces the evolution of learners' knowledge acquisition over time by explicitly modeling learners' learning and forgetting behaviors as well as the item difficulty. We model learners' learning and forgetting behaviors by taking account of their memory decay and the benefits of attempts on exercises, and propose a concept of cognitive item difficulty and a method to model this user-oriented difficulty adaptively in terms of the cognitive challenge it presents to different individuals. Empirical analyses were conducted to show the effectiveness of the KTM-DLF model and the impact of these factors on the learner knowledge assessment.

- A knowledge structure enhanced graph representation learning model for attentive learner knowledge assessment in Chapter 5. This approach solves both Issue 1 and Issue 2. It explores methods to infer the domain knowledge structure from the learner response data and integrates it with the original question−skill relation graph to enrich the data and alleviate the data sparseness. It proposes a knowledge structure enhanced graph representation learning model to learn the dense question and skill embeddings, and fuses these embeddings with other distinctive features to obtain the comprehensive question representation, thus alleviating the information loss of the existing skill-level models that neglect the distinctive information related to the questions themselves and their relations. Moreover, it solves the Issue 1 by discovering the knowledge structure and integrating it into the knowledge assessment process with other learning factors. Comprehensive evaluation results verified the superiority and interpretability of this approach in dynamically modeling the learning performance and discovering the knowledge structure from data.
- A knowledge interaction enhanced sequential modeling method for interpretable learner knowledge assessment. As shown in Chapter 6, we propose a novel model, called the *knowledge interaction-enhanced dynamic cognitive diagnostic assessment* (KIEDCDA), to dynamically trace the evolution of each learner's knowledge states during the exercise activities. It unifies the strength of the auxiliary memory capacity of the key-value memory network to enhance the representation of the knowledge state during learner performance modeling and the interpretability of the Item Response Theory (IRT) to explain the learner performance in terms of knowledge proficiency and item characteristics (i.e., item difficulty and discrimination). Moreover, we propose the knowledge interaction concept among knowledge concepts and incorporate it into the modeling procedure to further exploit the long-term dependencies in the exercising sequences, solving the Issue 1 to some extent. Based on these factors, this model can not only output the learners' knowledge proficiency in a multi-granularity manner but also output the item characteristics, making it possible to interpret the results and solve the Issue 3.

1.3 Outline of Thesis

The rest of this thesis is organized as follows:

Chapter 2 will describe the related work in three aspects: ITSs, learner and domain modeling in ITSs, and the learner knowledge assessment methods. The current development and limitations of the existing methods will be discussed.

Chapter 3 will introduce the datasets and the evaluation metrics used in this thesis.

Chapter 4 will explore the factors that influence the learning performance and the methods to quantify these factors and utilize them to model the dynamic evolution of learner knowledge. Specifically, it will introduce our first approach by modeling the dynamic knowledge construction procedure and cognitive item difficult.

Chapter 5 will introduce our second approach for learner knowledge assessment. It will show how the knowledge structure enhanced graph representation learning approach alleviate the data sparseness and the information loss when conducting the assessment.

Chapter 6 will show our third approach to obtain the fine-grained and interpretable results based on a deep learning model.

Chapter 7 will conclude this thesis. It replies to the three issues of DLKA, shows the remaining issues and explains the potential directions to improve learner knowledge assessment for ITSs.

2 Related Work

This chapter summarizes the models and techniques closely related with the task in this thesis, and gives further description of the research background. As this thesis is about the techniques that make the ITSs more intelligent, we first gives an overview of the intelligent tutoring systems in Section 2.1. Specifically, we conduct learner knowledge assessment by incorporating both learner and domain modeling. Section 2.2 discusses the techniques for learner and domain modeling in ITSs, respectively. Section 2.3 focuses on the research problem in this thesis and shows the related work in learner knowledge assessment from both static and dynamic perspectives. Based on the above analysis, Section 2.4 points out the limitations in existing methods and further introduces the positioning of this research in the relevant field.

2.1 Intelligent Tutoring Systems

From CAI to ITSs The employment of computers in education has a long history since its inception in the 1950s [57, 18] under the name of Computer Assisted Instruction

(CAI). The simple “linear programs” by Skinner [57] is regarded as the pioneer work in the field of CAI. The system presented a series of “’frame’ containing very simple problems to guide the students toward the desired goals. Students with different abilities, background, or prior knowledge, received exactly the same material in exactly the same sequence [18]. This system, together with many other earlier systems, are deemed to be unable to provide feedback and individualization, as they cannot obtain the knowledge of what they were teaching, who they were teaching or how to teach it.

During the late 1960s and early 1970s, the generative CAI systems came into the stage to improve the feedback and individualization. Uhr described a computer program that generates very simple questions in pre-defined formats in numerically oriented problem domain that are tailor made to student performance [58]. Wexler proposed a system which can dynamically generate simple instructional and remedial sequences that are used in non-numeric problem domains [59]. Although these systems drastically reduce the memory usage to store the learning materials by directly generating them, they are actually quite ad hoc and non of these systems has human-like knowledge of the subject they are tutoring [18].

To solve these issues, CAI systems have gradually integrated with artificial intelligence (AI) techniques and evolved into the advanced systems termed as “Intelligent Computer Assisted Instruction (ICAI)” [60]. In [60], Carbonel claimed that CAI systems could be more intelligent by incorporating AI techniques to overcome the existing issues. In recent decades, the term “Intelligent Tutoring System” (ITS) is used frequently as a replacement for ICAI [61]. In [18], Nwana thought that ITSs and ICAI are synonymous, as research under these two terms share the same intents and purposes. With the incorporation of AI in education, ITS has been a popular interdisciplinary research field across education, psychology, computer science, and cognitive science [16, 61]. From the perspective of pure research, the researches in ITSs will contribute to the discovery and test of more accurate theories of human learning; In practical level, the ITSs will facilitate one-to-one tutoring and provide supplement for the formal education, and the scalability and online characteristic also make the ITSs more affordable and convenient than face-to-face tutoring [17].

Researchers from multiple research fields have explored and solved issues in ITSs from different perspectives. A considerable consensus of the general architecture for the ITSs has been reached that ITSs consist of at least four basic components

[18, 61], as shown in in Figure 1.1. These four components are the domain model which contains the knowledge to be learned in a certain domain, the learner model which stores the profile of an individual learner, the pedagogical model which stores pedagogical knowledge and makes decisions about when and how to intervene, and the user interface that presents the learning interactions with learners.

Under this general architecture, many researches are conducted to improve the intelligence and adaptability of the ITSs. Among them, massive work is conducted on the topics of learner modeling and domain modeling (see Section 2.2), for example the learner knowledge assessment [50, 62, 63, 64, 65, 66, 28, 9, 29, 16, 30, 10, 23], learning style or preference detection [25, 67, 68, 69], cognitive engagement detection [24], affective states recognition [70, 71, 72], question-skill mapping or q-matrix learning [73, 74, 75, 76], knowledge graph construction [77, 78], and item analysis [79, 40, 80]. Pedagogical researchers have proposed various tutoring strategies for the ITSs to provide more adaptive services to learners, for example the adaptive navigation [53, 81], personalized pedagogical interventions [20], game-based learning strategies [82], dialogue-based tutoring [83].

To test the effectiveness of ITS on the learning outcomes, researchers have conducted a series of meta-analysis based on the existing studies. Steenbergen-Hu and Cooper conducted two meta-analysis of the effectiveness of ITSs on K-12 students' mathematical learning [84] and college students' academic learning [85], respectively. Twenty-six reports assessing the effectiveness of ITS on K-12 education settings and Thirty-five reports on higher education are analyzed. The findings demonstrated that ITS appear to have a more significant effect on college-level learners than on K-12 students [85]. For the college students, ITSs outperformed many instructional methods in a wide range of subjects, although they were not yet as effective as human tutors; For the K-12 students, ITSs overall appeared to have a small positive impact compared with regular class instruction, and they showed a greater positive impact on general students than on low achievers [84]. And they concluded that ITSs could be effective supplements to regular class instruction for students who are motivated and can self-regulate learning. Kulik and Fletcher [86] described a meta-analysis from 50 studies and showed that students learned using ITSs outperformed the counterparts from conventional classes in 92% of the studies, and the effect size was considered to be substantively great in 78% of the studies (above 0.66). The study by Ma et al [87] analyzed 107 published studies by comparing the outcomes of students learning from ITS and non-ITS environments. They found that students using ITSs had greater achievement compared with other settings (teacher-led, large-group instruction, non-ITS instruction) Moreover, no significant difference was observed between learning from ITS and learning from individualized human tutoring or small-group instruction, which verifies the rule of ITSs as relatively effective tools for learning.

Considering the effectiveness of ITS on the learning outcomes, this thesis explores the techniques that make the ITSs more intelligent and provide the adaptive services to maximize the learning gains.

2.2 Learner and Domain Modeling in ITSs

As we have mentioned in Section 1.1, learner and domain modeling are important for the ITSs to know who they teach and what they teach, providing the footing of penalization in ITSs. On the basis of these information, the ITSs link instructional materials structured in the domain model with the characteristics and needs of the learners [72]. In this section, we overview the related work on learner and domain modeling in ITSs, respectively.

2.2.1 Learner Modeling

Building an ITS with the ability to be adaptive and personalized is extremely challenging as learners usually have different needs and with different characteristics. A solution to this challenge is the technology of learner modeling [88]. Learner modeling is considered as the pillar of adaptive ITSs [72], which makes them superior to the “one-size-fits-all” tutoring in traditional web-based learning. It mainly undertakes two tasks: to infer the learner characteristics, and to represent them in order to be accessible by the ITSs for offering adaptation [88].

To construct a useful learner model, we need to first identify and select the aspects of the learners that influence their learning process and that should be included in the model, then build the model and maintain it up to date based on the long-term tracing of the learning activities in the ITSs. The literature review by Desmarais and Baker [16] revisited the learner model in ITSs before 2012, and discussed the

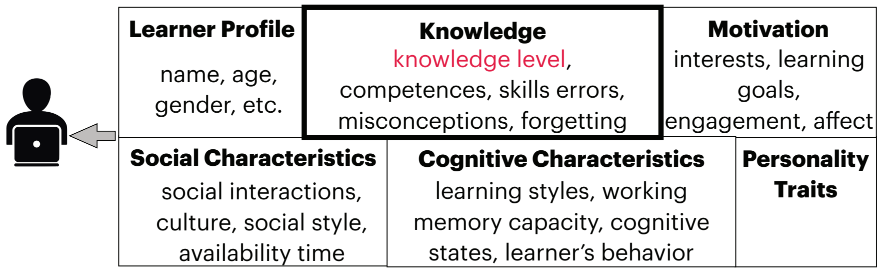

Learner Profile
name, age, gender, etc.
Knowledge
knowledge level, competences, skills errors, misconceptions, forgetting
Motivation
interests, learning goals, engagement, affect
Social Characteristics
social interactions, culture, social style, availability time
Cognitive Characteristics
learning styles, working memory capacity, cognitive states, learner's behavior
Personality Traits

Figure 2.1: Six categories of learner characteristics in existing learner modeling. Among them, learner knowledge is the most frequent and fundamental aspect for learner modeling.

advancements of learner modeling from learner knowledge-based modeling to the modeling of other key constructs, such as learner motivation, emotional and attentional state, meta-cognition and self-regulated learning, etc. Chrysafiadi and Virvou [88] reviewed learners' characteristics that should be considered in the learner modeling and the approaches for and potential use of learner modeling. Pelánek [23] reviewed the techniques for learner modeling from a broad aspect, including both knowledge model and domain model. A more recent survey by Abyaa et al. [72] contributed to the identification of the learners' individual characteristics and described the most used techniques for modeling them. They divided the learner characteristics in six categories: the static learner profile (such as the age, gender, name and other demographic information), knowledge, cognitive characteristics, social characteristics, motivation and personality, as shown in Figure 2.1.

Various researches on learner modeling have been conducted from different aspects and provided different kinds of adaptation to facilitate the ITSs. Learner knowledge is assessed in a learner model to uncover a learner's strength and weakness of knowledge to further deliver the most appropriate learning materials and feedback. Researches in [50, 62] estimated learner knowledge at certain time point through tests; while some others [28, 9, 29, 30, 10] dynamically traced learners' evolving knowledge states in the ITSs using the massive learning logs. Cognitive characteristics, such as the learning styles, facilitate the ITSs to make decisions about the most effective learning strategies [88]. Two popular learning style models, VARK (Visual, Aural, Read/write, and Kinesthetic) [67] and FSLS (Felder–Silverman learning style: active/reflective, sensing/intuitive, visual/verbal, and sequential/global) [68] are widely adopted in existing ITSs. Binh et al. [69] built an ITS and adopted the FSLS model to identify learning styles, based on which the lessons were chosen to satisfy learners' need. Their results revealed some advantages over the traditional class in various ways. Learner motivation, such as the affective state and engagement, is an important consideration for the ITSs when choosing the proper learning methods that increase the effectiveness of interactions. Affective states, such as interested, frustrated, bored, distracted, focused and confused, are found to be highly related with the learners' motivation [88], and some of them may lead to failure interaction with the systems, such as gaming the system and off-task behaviors [89]. Other characteristics of learners, such as the social interaction and personality, are also incorporated in previous researches [72, 88], providing additional information for the ITSs to provide corresponding tutoring services.

Ideally, a learner model should incorporate all aspects of learner characteristics that may have an impact on his/her learning. However, it is not only non-trivial but maybe impossible to build such a comprehensive model [18], as inferring different aspects of learner characteristics requires different channels of the input data, and it is quite tough to obtain the multi-modal data simultaneously. Hence most of the existing ITSs only incorporate part of the learner characteristics to build the learner model. Among the various learner characteristics that contained in the the learner models, knowledge of learners is the most frequent and fundamental aspect for learner modeling, and it is also the primary model type utilized in the majority of existing ITSs [23]. This thesis follows this paradigm, and aims to assess learner knowledge for building the learner models. We will focus on this topic and further describe the related work in detail in Section 2.3.

2.2.2 Domain Modeling

Domain model, which models the content of the learning domain, provides the ITS with the knowledge of what they are teaching [61]. Pelánek [40] gave a more abstract definition of this concept as "designing an appropriate organization of individual learning objects to higher-level units and specification of relations among these units"; while in practical development of a usable ITS, domain modeling is usually conducted by managing the learning objects to make them in well-defined organization. It is crucial in developing the ITSs, and can be used in many ways, such as personalization of learning objects for learners, feedback of the learning progress, organization of the content [40].

For a specific learning domain, there are generally a set of *knowledge components (KCs)* to learn, KC is also termed as skill, attribute, knowledge and concept synonymously. These KCs are usually embedded in a series of learning objects, such as presentation, exercises and learning videos. In this thesis, we denote learning objects by the generic and commonly used term *items*. Actually, exercises are the widely used materials in the ITSs, hence items are specifically defined as exercises in many of the existing studies. For the domain modeling, the main task is to elicit these KCs and map between items and KCs, assess the characteristics of items, and find the relation among KCs and items.

KC Elicitation and Item-KC Mapping KCs in a domain are treated as"organizational units that group together related items" [40]. These KCs can be facts (e.g., one-digit multiplication) or rules (e.g., solving equation) that represent the knowledge to be learned in the domain. In most ITS, they are elicited by experts in a manual manner. This is a highly time-consuming process, particularly for a complicated subject with a large quantity of knowledge [18]. Nevertheless, there is severe issue of consistency as trade-offs should be made between granularity and coverage. To cope with this, researchers have explored methods to automatically elicit KCs from the items using natural language processing techniques. Chau et al. [75] proposed to use automatic key-phrase extraction to obtain key-phrases and use them as KCs to index each textbook section. This method is especially suit for the textbook-based learning, where the items are mainly reading materials. However, for the items like exercises, this method has rarely explored as exercises usually have very short text and the KCs to solve a specific exercise are implicit and usually cannot be directly obtained from the text.

Given the KCs in the domain, they are usually embedded in and practiced by items, and naturally a mapping is built between these KCs and items. This mapping is represented by a Q-matrix, as shown in the example of Figure 2.2, and can be represented by a bipartite graph. Generally, items are designed for the aim of practicing the KCs, hence in most of the ITSs, the Q-matrix is designed by the experts and stored

		KC			
		Add	Subtract	Multiply	Divide
Item	3+5	1	0	0	0
	345-246	0	1	0	0
	2.9+5.6*6.9	1	0	1	0
	25/4-4.75	0	1	0	1

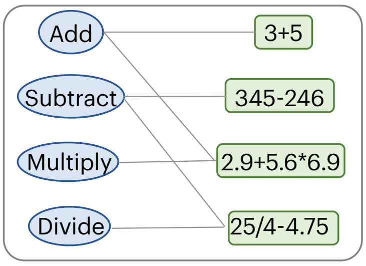


Figure 2.2: Q-matrix and the item-KC mapping graph.

in the backends of the systems. To improve the scalability of the ITS to incorporate the overwhelmingly large numbers of items emerging on the Internet, many researches have been conducted to learn the Q-matrix automatically from the learning data. Liu et al. [76] introduced an estimator of the Q-matrix under the setting of the DINA model in a data-driven manner. Sun [66] proposed a recursive method that updates the q-matrix based on the Boolean matrix factorization. However, these methods obtain the q-matrix with unknown KCs, thus making them difficult to interpret as expert-made and the inferred q-matrices do not often coincide. To improve the interpretability, Matsuda et al. [90] exploited both student performance data and the text of items to build a statistical model to infer the q-matrix, the bag-of-words (BoW) strategy used for text analysis provided some kind of explanation of the KCs. Some researchers refine the existing q-matrix to make it more fit to the learner performance data [91, 92]. In this thesis, as the focused topic is learner knowledge assessment, we also assume that KCs in the domain are given and the q-matrix is already labeled by the experts following most of the existing studies.

Item Assessment Item assessment is to obtain the characteristics of items, based on which they are structured and managed. Items, even contain the same KCs, are generally designed with different properties, such as difficulty, discrimination, quality, complexity and similarity with others. Obtaining these properties of items is useful for the personalization of ITSs, for example, providing a learner with the proper challenging items that suitable for his/her ability.

Pelánek [80] gave an overview of the complexity and difficulty of items in ITSs, he presented a simple distinction between complexity and difficulty measures that complexity is based only on item description while difficulty is based only on data about student performance. He concluded from the existing work that complexity is usually measured by the length of text, the number of KCs in an item, and the number of steps required to solve an item; while measurements of difficulty are dependent on learners' performance, such as failure rate and median response times. Minn et al. defined the difficulty of an item as the ratio of the number of failed attempts to the total number of attempts by the set of students in a system who have attempted the same item [93]. Wang et al. [94] modeled learners' exercising logs on a large number of questions and predicted their potential answer to unseen questions to impute the missing value in the response matrix. Based on the completed matrix, they quantified the difficulty of questions by the incorrect rate, and defined the quality of questions using the information gain of estimating learner ability conditioning on the answer to questions.

The item response theory (IRT) model [50] is the widely studied model in Psychometrics for modeling the probability that a learner answers item correctly based on his/her ability and the item difficulty and discrimination. By fitting on the reponse data, it infers the item difficulty and discrimination. Similar models that can also be used for item assessment are the additive factor model (AFM) [95] and Performance factor analysis (PFA) [96]. Based on IRT model, many researches on item assessment have been conducted. Pankiewicz et al. [97] compared four methods of item difficulty estimation: learner feedback, incorrect rate, and Elo and Glicko based rating algorithms with reference values provided by the Item Response Theory model. Highest correlation has been found for the Glicko algorithm. Similar research is also conducted in [98]. Ayers and Junker [99] used Item Response Theory based model to predict item difficulty based on the number of skills required for the items. They assumed that the more skills required for each item, the more difficult the item is expected to be.

Some researchers extract the feature from the items and train transfer models to estimate the item characteristics. Benedetto et al. [100] proposed a framework for assessing the difficulty and the discrimination of newly generated multiple-choice questions using natural language process paradigm. They used the Item Response Theory model to obtain the ground-truth difficulty and discrimination of learner-attempted exercises, and then trained regressors based on the NLP features of questions and the ground-truth. Fang et al. [101] predicted the difficulty of visual-textual exercise using a multimodal embedding extractor to obtain a unified representation for exercises, and then training a classifier to predict the difficulty in a supervised manner.

To facilitate the item recommendation in ITSs, researchers have proposed various methods to measure the item similarity. Pelánek [102] provided a overview of approaches for quantifying similarity of items from diverse domains in two categories: similarity based on item statements, metadata and solutions and similarity based on performance data. Rihák and Pelánek [103] compared different measures of item similarity and showed that Pearson correlation is a good similarity metric based on learner performance data and the additional response times increase stability of the measures on small data. Nazaretsky et al. [104] proposed an item-similarity measure termed Kappa learning based on learner performance data that can capture similarity in the context of learning. Mussack et al. [105] discovered item similarity through combining item features and user behavior using a deep learning method. Liu et al. [106] proposed a deep learning framework for finding similar exercises by learning a unified semantic representation from the heterogeneous item data (i.e., texts and images).

Knowledge Graph Construction In education domain, pedagogical concepts usually have various relations with each other, such as the prerequisite relations. The prerequisites between these concepts can be represented as a knowledge graph [53, 107, 108]

Knowledge graph is usually designed by experts in a certain domain, and building this graph is quite labor-intensive work, especially in the case of massive concepts. With the knowledge graph, ITSs can provid personalized learning paths and services to accommodate the needs of different learners. Automating this process has been attempted in several studies. Most of the existing methods for knowledge graph construction identify the latent skills required for answering the questions, and find the similarities among the questions in the domain for clustering the potential knowledge graph. Pan et al. [109] automatically inferred the prerequisite relation between knowledge concepts in MOOCs using natural language processing techniques. Wang et al. [110] proposed a latent-variable selection method with regularization for cognitive diagnostics, which learns the skill hierarchies from learner response data. Using a DKT model, Zhang et al. [77] discovered the topological order of skills from learners' exercise performance. Chen et al. [78] adopted named entity recognition

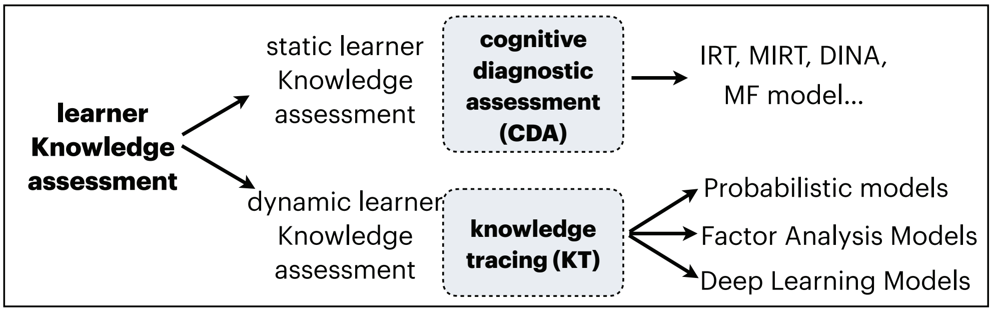

learner Knowledge assessment
static learner Knowledge assessment
cognitive diagnostic assessment (CDA)
IRT, MIRT, DINA, MF model...
dynamic learner Knowledge assessment
knowledge tracing (KT)
Probabilistic models
Factor Analysis Models
Deep Learning Models

Figure 2.3: Static and dynamic learner knowledge assessment.

techniques to extract educational concepts from item text, and utilized association rule mining on the performance data to identify prerequisite relations among these concepts to build the Knowledge graph.

2.3 Learner Knowledge Assessment

As described in Section 1.2, learner knowledge assessment is to obtain learner knowledge states based on the learners' explicit learning feedback on various types of learning materials. Some researchers conducted learner knowledge assessment based on the learners' interaction results on gradable types, such as assignments and exercises [9, 10, 11], while some others solved this task utilizing the non-gradable types, such as readings [44] and video lectures [45]. Another direction of work is based on the multiple learning resource types concurrently [46, 47, 48]. In this thesis, we focus on the learner knowledge assessment by utilizing their performance on exercises, which can be interpreted as an explicit indication of their implicit knowledge. Based on the application context, existing work can be divided into two categories: static learner knowledge assessment for *testing* and dynamic learner knowledge assessment for *learning*, as shown in Figure 2.3. For the testing context, learner knowledge assessment is to obtain the fine-grained diagnostic reports on learner knowledge instead of just the ranks or final scores. It is also termed as *Cognitive Diagnostic Assessment* (CDA). The data for analysis is the learners' performance data on a single summative quiz/test with limited items, such as the GRE (Graduate Record Examinations). For the learning context, learner knowledge assessment is to obtain the learners' long-term evolving knowledge states for the purpose of providing adaptive tutoring. This category is known as *Knowledge Tracing* (KT), and the input data is generally the learners' long-term exercising logs in the systems.

2.3.1 Static Learner Knowledge Assessment/CDA

CDA is usually conducted after a summative test, aiming to measure learners' knowledge states on a set of KCs through some diagnostic assessments. This topic has been widely explored by researchers in the fields of psychometrics and data mining because of the fundamental value of the diagnostic results for both instructors and learners to assess progress towards attaining learning objectives.

Psychometrists discover learners' latent knowledge proficiencies by designing delicate psychological models. These models generally consider the learners' personal traits and item characteristics, which makes them interpretable. IRT [56] and DINA [62] are two of the most renowned models in this category. IRT diagnoses a learner's knowledge proficiency using a unidimensional variable (i.e., latent trait), which can be seen as a general level of KC attainment. It uses an interaction function to model the probability of a learner's correct solving of an item based on his latent trait θ and item characteristics (i.e., item difficulty and discrimination). The widely used two-parameter logistic IRT model [56] is described as follows:

$$p(\theta) = \frac{1}{1 + e^{-Da(\theta - b)}} \tag{2.1}$$

where a and b depict two parameters for each attempted item denoting item difficulty and item discrimination, respectively; $p(\theta)$ is the correct probability; and D is a constant usually set to 1.7. The DINA model [62] describes each learner's proficiency level using a binary vector, where one represents mastering of a specific KC and zero otherwise. Unlike IRT, the DINA model must infer this multidimensional vector by using the Q-matrix. Accordingly, researchers have extended IRT-related models and proposed MIRT models [111] to indicate more complex and diverse student latent attributes by representing the learner's latent proficiency by a vector.

Meanwhile, data mining researchers have proposed various data-driven methods to assess the learners' proficiencies based on Matrix factorization (MF) [64, 65, 66]. Nguyen et al. [64] used multi-relational MF to diagnose the learner proficiency in

ITSs by considering the interaction between the learners and items and exploiting the possible relation between learners and the items for improving the prediction accuracy. Sahebi et al. [65] proposed a tensor factorization-based approach to model the increases in learners' knowledge by using a feedback-based constraint on the previous proficiency and the current item. Sun et al. [66] attempted to use the Boolean MF method to express conjunctive models in CDA and automatically learn the knowledge state matrix from the learners' item response matrix.

The psychological model based approaches provide some explainable results for learner knowledge, while the MF-based approaches are long criticized by their unexplainability. Moreover, in practical scenarios, the number of KCs for setting the psychological models must not be too large so as to be statistically supportable [112, 113], which makes the assessments of a large number of KCs impractical [114], especially in large-scale adaptive learning environments. Conversely, IRT-based assessments provide coarse-grained uni- or low-dimensional values to represent the general proficiency of learners, which may not directly represent their strengths and weaknesses. Despite this limitation, IRT-based models have been widely used in practical assessment because of their interpretability and simplicity. However, all these studies perform CDA under static assumption (i.e., infer learner proficiency in independent assessments at some time points); therefore, the temporal factor for the learner proficiency evolution is greatly ignored, which makes these approaches unable to be directly adopted into the dynamic learning context in ITSs.

2.3.2 Dynamic Learner Knowledge Assessment/KT

To utilize the temporal factor of learning in ITSs, various models have been proposed to dynamically model the learner performance from a long period and trace the learners' knowledge over time [63, 10, 32, 31, 27, 30]. The existing KT methods can be generally divided into three main categories: probabilistic models, factor analysis models and the deep learning models [11].

Probabilistic Models Bayesian knowledge tracing (BKT) [9] is a pioneer model for the task of KT. It is a probabilistic model based on the hidden Markov model (HMM) that separately tracks the proficiency of each KC based on the exercising logs, i.e., each KC has a specific BKT model. There are four parameters in BKT: transition probability $P(T)$, the initial probability of mastery $P(L_0)$, slip probability $P(S)$ and guess probability $P(G)$. $P(T)$ represents the probability of transition from the non-mastered to mastered state, $P(S)$ is the probability that a learner will incorrectly answer an item in spite of mastery, and $P(G)$ is the probability that a learner will guess correctly an item in spite of non-mastery. Given the exercising logs on a specific KC, the estimation of learner knowledge state and the probability of correct answer are as follows:

$$P\left(L_t\right)=P\left(L_t \mid \text{ Answer }\right)+\left(1-P\left(L_t \mid \text{Answer}\right)\right) P(T), \tag{2.2}$$

$$P\left(C_{t+1}\right)=P\left(L_t\right)(1-P(S))+\left(1-P\left(L_t\right)\right) P(G), \tag{2.3}$$

where $P(L_t)$ is the probability that the learner has mastered the KC at timestamp t, and $P(C_{t+1})$ is the prediction of correctly answering the next item based on the current knowledge state. The posterior probability $P(L_t|Answer)$ is estimated using the Bayesian formula based on the correct or incorrect answer to the previous item.

$$P\left(L_t \mid \text{ correct }\right)=\frac{P\left(L_{t-1}\right)(1-P(S))}{P\left(L_{t-1}\right)(1-P(S))+\left(1-P\left(L_{t-1}\right)\right) P(G)} \tag{2.4}$$

$$P\left(L_t \mid \text{ incorrect }\right)=\frac{P\left(L_{t-1}\right) P(S)}{P\left(L_{t-1}\right) P(S)+\left(1-P\left(L_{t-1}\right)\right)(1-P(G))} \tag{2.5}$$

Baker et al. [63] extended BKT by contextually estimating the probability of slip and guess and alleviated the model degeneracy. BKT-based models is regarded to have information loss in the modeling process as they do not consider the contextual trial sequence of all skills and inter-skill similarity.

Factor Analysis Models To trace a learner's proficiency using the whole long-term contextual sequence, researchers have proposed various factor-analysis models. These models pre-design a delicate model framework and assign the considered factors as model parameters, which are learned from the data to generalize the observations.

The IRT model [50] has been simplified and used as regression model for modeling the learner performance $p_{i,j}$ dynamically based on learner ability α_i and the item difficulty d_j. The one-parameter dynamic IRT is as follows:

$$logit\ p_{i,j} = \alpha_i - d_j \tag{2.6}$$

Multidimensional IRT (MIRT) [115] extends IRT by considering the interactions of the multidimensional embedding vectors of the two variables in Eq. 2.6.

The additive factor model (AFM) models the probability of attempting an item correctly by considering the difficulty of the KCs involved in the item and the number of attempts on items that require the involved KCs [95]. Performance factor analysis (PFA) improves the AFM by considering separately successful and failed attempts [96]. The probability of an item j being successfully answered by student i is defined as follows:

$$logit\ p_{i,j} = \Sigma_{k \in KC(j)}(\beta_k + \gamma_k W_{ik} + \delta_k F_{ik}) \tag{2.7}$$

where β_k is the difficulty parameter for KC k involved in item j, and W_{ik} and F_{ik} are the numbers of successful and failed attempts, respectively, required for KC k.

More recently, Vie and colleagues have proposed a knowledge tracing machine (KTM) based on factorization machine (FM) [116], a generic framework that incorporates side information (e.g., users, items, skills, win and fail attempts) into the student model [30]. They modelled the probability to observe a positive outcome as follows:

$$p(y_t = 1) = \sigma(\mu + \sum_{i=1}^{N} w_i x_{i,t} + \sum_{1 \leq i < j \leq N} \langle v_i, v_j \rangle x_{i,t} x_{j,t}) \tag{2.8}$$

where μ is a global bias, and $x_{i,t}$ and $x_{j,t}$ are the i_{th} and j_{th} abstract features in a vector of totally N features collected at time t. w_i is the bias of feature i and $v_i \in R^{dim}$ its embedding in dim dimension. In KTM, the features in an input sample are generally sparse features including: which user attempted which item, the KCs involved in the item, and the win/fail information related with previous attempts, etc. The first two terms in Eq. 2.8 are actually regression terms and the last term models the pairwise interactions between the high-dimensional embeddings of features in the input, which allows high quality parameter estimates of higher-order interactions under sparsity [116]. A sample is generally encoded into a vector of sparse features by concatenating all the features in the one-hot encoding manner. They have proven that the KTM model encompasses several EDM models, including IRT, MIRT, AFM, and PFA [30]. Wang et al. adopted variational inference to perform Bayesian inference for factor analysis models and output the uncertainty of model estimation [117].

Deep Learning Models Various neural networks have been recently used for the KT task and show significant improvements in model performance over traditional models due to their excellent abilities in conducting learning on big data [10, 32, 118].

The pioneering work on deep KT (DKT) [10] obtained the learners' latent knowledge proficiency, which was extracted from the exercising sequences by a recurrent neural network (RNN) or long short-term memory (LSTM). Through learning from the input sequences of learners' learning history, the hidden layer retains relevant information that is useful for the future performance prediction, and hence the hidden state in the RNNs can be intuitively seen as embedding the knowledge states of students [119].

Nagatani et al. [120] extended the DKT model to enable it to predict learners' future performance by considering their forgetting behavior. In addition to encoding learners' attempts (trials and accuracy), it also incorporates time- and count-based side information to model learners' forgetting behavior, thereby showing that the inclusion of forgetting information results in performance improvement. DKT-DSC (Deep Knowledge Tracing with Dynamic Student Classification) [121] improved the DKT model by capturing learners' learning ability through assigning learners into distinct groups with similar ability at regular time intervals dynamically. By incorporating this side information (label of group level) with student trial sequence, it improves performance significantly as compared with the DKT model. Similar to the former two methods, DKT-DSC can be used only for problems with a single associated knowledge component.

Although DKT-based approach has gained significant performance improvement, it is limited in terms of pinpointing the learners' actual proficiency on specific KCs. To overcome this limitation, DKVMN [32] was proposed using an auxiliary memory to record the proficiency of each latent KC based on a memory-augmented neural network (MANN). Other related models have also been proposed based on this model [31, 122, 123, 124]. Sequential key-value memory networks (SKVMNs) [31] modeled student learning by unifying the strengths of RNN (recurrent modeling capacity) and MANN (high memory capacity).

A newly proposed network named Transformer [125] has also been adapted to learner knowledge assessment in various ways [118]. To focus on the relevant previous interactions, the Transformer framework applies a self-attention mechanism to the input data, and hence incorporates the inner relations in the exercise sequences into the network. Attention based KT models inspired by this work have become an active research area [126, 127, 1]. Graph-based KT models [42, 41, 51, 128] have become prevalent in recent years, but the focus is different from the above deep learning models, as they main solve the embedding representation in the KT task, we will describe this branch of researches in the following paragraph.

Some researchers have questioned the using of deep learning in the educational setting [29], since deep learning does not appear to be the panacea, particularly when an explicit underlying theory and interpretability matter [129]. Recent work has also showed that Bayesian extensions of IRT [130] and extended BKT [119] outperformed or performed just as well as neural networks for proficiency estimation. Nevertheless, deep learning have widely used for solving this task and have obtained significant improvement over other models, and some models have already been applied into the real-world ITSs (e.g., the Riiid TUTOR [1]).

Graph-based models are a special branch of deep learning models, here we describe such models separately.

Graph-based Models The relations between questions and skills have been considered in various graph-based KT models [42, 41, 51, 128]. Liu et al. [42] built a question–skill bipartite graph based on the Q-matrix in the domain and pre-trained the question and skill embedding in the graph using three constraints of explicit question–skill relations and implicit relations of skill similarity and question similarity. As the dense embeddings of questions and skills contain the relations in the graph, the pre-trained embedding-fed KT model outperforms the baseline models. Yang et al. [41] incorporated the question–skill correlations via embedding propagation on the question–skill relation graph using a graph convolutional network (GCN). Tong et al. [51] introduced problem schema with a hierarchical exercise graph to KT. The problem schema clustered similar exercises into the same group to incorporate the question–question relations. Pandey et al. [127] calculated the exercise–exercise

[1] https://riiid.com/en/product

relations based on the performance data and the exercise text, and incorporated it into a transformer model for relation-aware KT. All of these methods exploit the relation information between questions and skills in various graphs and introduce the distinctive information of questions into the KT task. Some other researchers utilized the skill-skill relations into the KT procedure. Nakagawa et al. [128] and Chen et al. [107] incorporated the knowledge structure information into the KT procedure. Chen et al.'s work [107] assumes that the knowledge structure is already given by experts, and models the prerequisite as an ordering pair. In this way, the mastery of related skills is constrained by referring to the knowledge structure. Nakagawa et al. [128] used the knowledge structure information and a graph neural network (GNN) to update the learners' hidden knowledge states.

Besides the main three categories of KT models, there are some variants with additional information considered in the modeling process.

KT with Learning and Forgetting Most of the above-mentioned KT approaches model students' learning in an implicit manner by obtaining their (implicit) knowledge states through learning from sequences of multiple attempts. However, there are only a few studies in the field of KT that have addressed learning and forgetting explicitly and simultaneously [120, 131, 132, 29, 28, 133], while either simplifying the forgetting behavior or just ignoring it.

Chen et al. [28] embedded students' learning and forgetting as a prior and designed a probabilistic matrix factorization framework by incorporating this prior for tracking student knowledge proficiency. Mohammad et al. introduced a forgetting parameter, and counted the number of intervening trials and treat each as an independent chance for forgetting some skill. By incorporating this forgetting factor into the classic BKT model, they proved that it has the potential to be sensitive to interspersed trails in the trial sequences and outperforms the ordinary BKT [119]. Nagatani et al. [120] explicitly incorporated information related to forgetting into the DKT framework. It models learning and forgetting by considering the number of times a student has attempted an item and the lag time from the previous interaction with the same item. However, this DKT-based method cannot be applied to items with multiple KCs. DAS3H (item Difficulty, student Ability, Skill, and Student Skill practice History) [29] builds on the DASH (Difficulty, Ability, and Student History) model [132] and uses KTMs [30] to handle multiple KC tagging. It depends on the temporal distribution and the outcomes of past practices to simulate memory strengths and estimates the difficulty parameters for each item and the skills contained. The model is defined as follows:

$$p(y_{i,j,t} = 1) = \sigma(\alpha_i - \delta_j + \sum_{k \in KC(j)} \beta_k + h_\theta(a_{i,s,1:t})) \tag{2.9}$$

$$\begin{aligned} h_\theta(a_{i,s,1:t}) = \sum_{k \in KC(j)} \sum_{w=1}^{W} & \theta_{k,2w+1} \, log(1 + c_{i,k,w}) \\ & - \theta_{k,2w+2} \, log(1 + a_{i,k,w}) \end{aligned} \tag{2.10}$$

where the probability of student i correctly attempting item j at time t depends on his ability α_i, the difficulty of the item δ_j, and the sum of the difficulty values β_k of the KCs involved in item j, as well as the synthesized result of learning and forgetting $h_\theta(a_{i,s,1:t})$. In h_θ, w is the index of the time window before the time t, and $c_{i,k,w}$ is the number of times that KC k has been correctly recalled in window w in $a_{i,k,w}$ times of attempts; intuitively, h_θ can be viewed as memory strengths synthesized by learning and forgetting.

KT with Item Difficulty Modeling Several studies have already attempted to incorporate the item difficulty in KT, and the experimental results showed empirically the benefits of adding this difficulty information for this task [93, 16, 134, 30, 135]. These studies generally model the difficulty of an item as a notion or a function of KCs associated with the item.

Pardos et al. captured the difficulty of items belonging to a particular skill being fit by individualizing the guess and slip parameter of each item and integrated it into the BKT model [134]. Difficulty is actually a function of KCs mapped to items (in the parlance of their paper, difficulty is mapped to items belonging to a specific skill). The variations of IRT models also incorporate a parameter indicating the difficulty of an item, which is a notion specific to IRT [135]. Moreover, the KTM model learns a vector of parameters on the one-hot encoded item vectors to obtain the difficulty coefficients of all of the items [30]. Minn et al. defined the difficulty of an item as the ratio of the number of failed attempts to the total number of attempts by the set of students in a system who have attempted the same item [93]. Unlike the above models, the

DAS3H model estimates the difficulty parameters for each item and also for all the skills they involve. They do not assume that items with same skill are interchangeable, and believe that their difficulties may differ from one another [29]. Therefore, difficulty parameters are calculated both in item and also in skill level.

KT with Knowledge Interaction The knowledge interaction models the dependencies among KCs, where the opportunities of practicing some KCs in the previous attempts affect the knowledge proficiency of later KCs in the exercising sequences. The interdependencies among KCs have long been explored as knowledge graphs or maps [107, 108, 53], where KCs are represented as nodes and the prerequisite and subsequent relations between them are described as the link between them. Using knowledge interaction matrix is another form of modeling the relation between KCs. Chen et al. [107] exploited the prerequisites as constraints on student proficiency prediction and incorporated them into the DKT framework. Although showing some performance improvement, the method utilized a manually labeled prerequisite matrix in model testing, which is a labor-intensive task in a large-scale assessment. Nakagawa et al. [128] learned the latent knowledge structure information and used a graph neural network (GNN) to update the learners' hidden knowledge states. Some researchers utilized the relations between items to indirectly model the knowledge interaction among KCs, as KCs are embedded in the items. Liu et al. [1] proposed the exercise-aware KT model that leveraging the exercise text to enhance the KT process. Exercise similarity calculated on the exercise representations was used for the attention calculation to model the indirect knowledge interaction. Pandey et al. [127] calculated the exercise–exercise relations based on the performance data and the exercise text, and used it as the attention constraint for relation-aware KT. A similar idea is also presented in [51].

2.4 Limitations of Existing Methods

By analyzing the existing researches on DLKA task, we find three main limitations, which motivated the researches in this thesis.

Insufficient Learning Factor Modeling The knowledge construction procedure is constantly evolving because learners in ITSs dynamically learn and forget over time. Cognitive psychology has long verified that the knowledge acquisition procedure is not only related to the learners but also the learning materials in the domain. To model this complex procedure, many factors need to be considered to make the model accurately assess learners' real knowledge. Unfortunately, to the best of our knowledge, most of existing approaches consider only a fragment of the information during the learning process that results in the change of learner knowledge acquisition, and the problem of making use of rich information during learners' learning interactions to achieve more precise prediction of learner performance in KT remains under-explored. Hence we must first pinpoint the factors that influence the evolving knowledge and propose methods to quantify them to more precisely assess the learner knowledge.

Data Sparseness and Information Loss The performances of existing approaches greatly suffer from the sparseness of the input data and the information loss when modeling the learning process. In the ITSs, there are usually a large number of items with limited number of KCs, and each item is only related with very small number of KCs. The high scarcity and the large quantity of items present great challenges to the DLKA task as each learners in the ITSs generally answers just a small proportion of potentially non-overlapping items. The adequacy of DLKA is still challenged by the sparseness of the learners' exercise data. To alleviate the sparseness problem, most of the exiting studies are performed at the skill-level rather than the question-level, as questions are often numerous and associated with much fewer skills. However, at the skill level, KT does not distinguish questions containing the same skills and hence neglects the distinctive information related to the questions themselves and their relations. Moreover, almost all of these models simply assume that all questions and skills are independent, which is unrealistic in the actual learning process. In this case, the models can imprecisely infer the learners' knowledge states and might fail to capture the long-term dependencies in the exercising sequences.

Fine-grained Assessment and Interpretability Deep learning models have obtained excellent results to model the learning process by leveraging the powerful representation ability of the deep neural networks in a data-driven manner. However, deep neural network is regarded as a black-box, and most of the deep learning models, especially the RNN based sequential models, retain the learner knowledge in a hidden vector or model parameters. This works well for the prediction of learners' future performance, but from the perspective of proving good tutoring services to learners, their fine-grained knowledge proficiencies in a multi-granularity manner are particularly important. Moreover, these methods found it difficult to go deeper into the explanation of the learners' performances in terms of their current knowledge proficiencies and item characteristics. Hence how to track and explain learners' evolving knowledge states simultaneously remains to be an important issue.

3 Data Collection and Evaluation Metrics

This chapter describes the datasets and the evaluation metrics to measure the model performance in this thesis. These datasets are public data collected from several real-world ITSs, we first give a short introduction to these ITSs in Section 3.1 and then describe these datasets in detail in Section 3.2. Section 3.3 introduces the evaluation metrics for performance measurement.

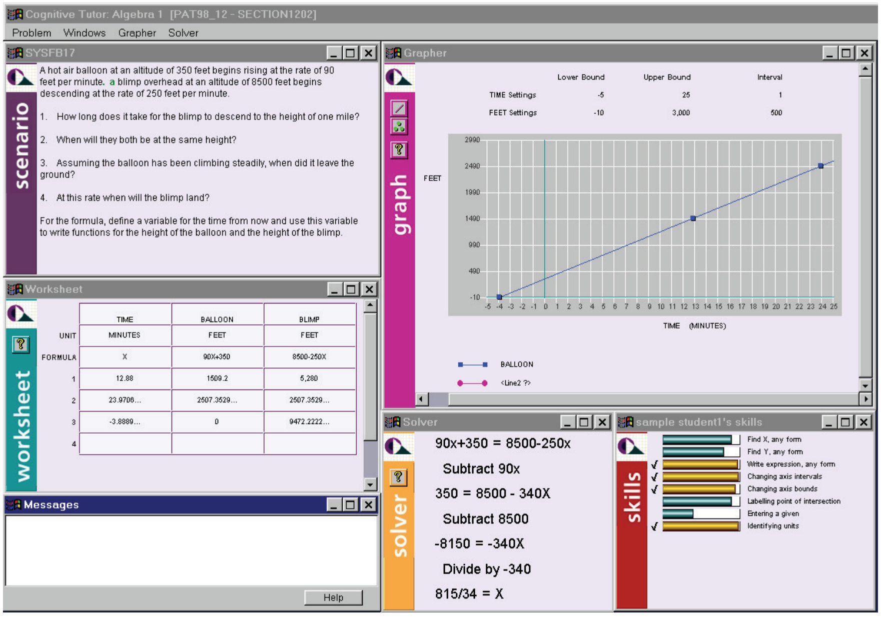
3.1 Real-word ITSs for Data Collection

In this thesis, we use several public datasets collected from the real-world ITSs, e.g., Carnegie Learning's Cognitive Tutor (CT) [1], the ASSISTment system [2], and Riiid's Santa TOEIC [3]. To gain a better understanding of the data, in this section we give a short introduction to these three main ITSs.

[1] https://www.carnegielearning.com/solutions/math/

[2] https://new.assistments.org/

[3] https://riiid.com/en/product



A hot air balloon at an altitude of 350 feet begins rising at the rate of 90 feet per minute. a blimp overhead at an altitude of 8500 feet begins descending at the rate of 250 feet per minute.

1. How long does it take for the blimp to descend to the height of one mile?

2. When will they both be at the same height?

3. Assuming the balloon has been climbing steadily, when did it leave the ground?

4. At this rate when will the blimp land?

For the formula, define a variable for the time from now and use this variable to write functions for the height of the balloon and the height of the blimp.

	TIME	BALLOON	BLIMP
UNIT	MINUTES	FEET	FEET
FORMULA	X	90X+350	8500-250X
1	12.88	1509.2	5,280
2	23.9706...	2507.3529...	2507.3529...
3	-3.8889...	0	9472.2222...
4			

	Lower Bound	Upper Bound	Interval
TIME Settings	-5	25	1
FEET Settings	-10	3,000	500

90x+350 = 8500-250x

Subtract 90x

350 = 8500 - 340X

Subtract 8500

-8150 = -340X

Divide by -340

815/34 = X

Find X, any form

Find Y, any form

✓ Write expression, any form

✓ Changing axis intervals

✓ Changing axis bounds

Labelling point of intersection

Entering a given

✓ Identifying units

Figure 3.1: The interface of Cognitive Tutor for Algebra I [3].

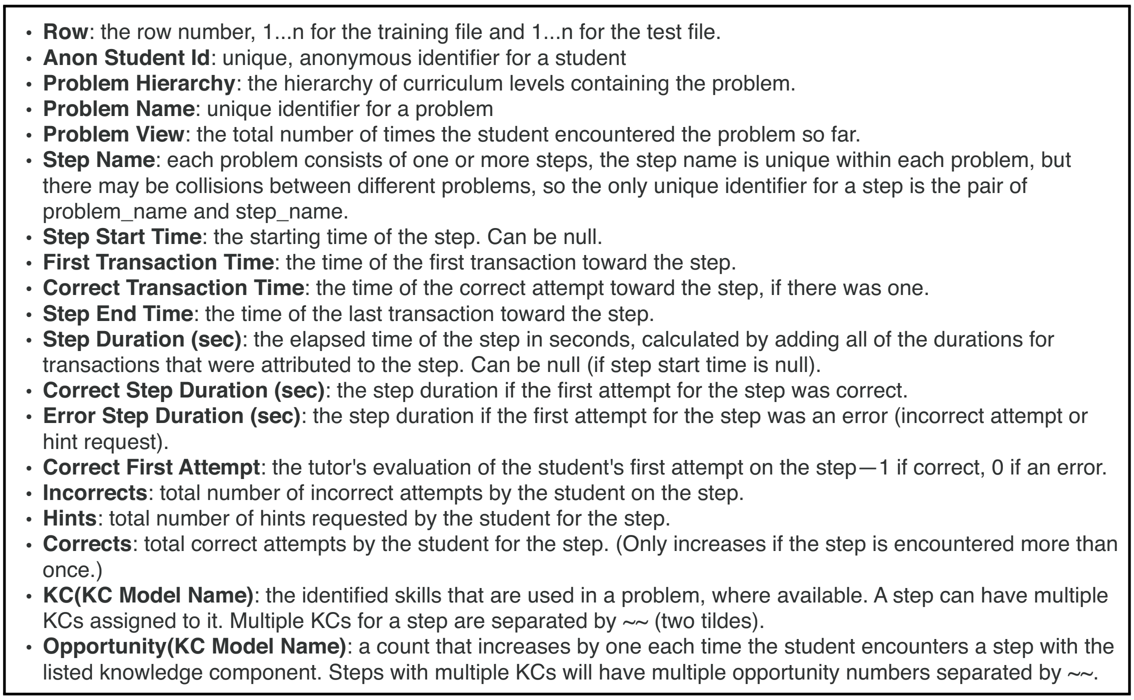
3.1.1 Cognitive Tutor

CT is a mathematics ITS created and supplied by Carnegie Learning, an enterprise formed by scientists at Carnegie Mellon University, and now has developed into the MATHia program in K-12 education and Mika in higher education. It is also the first successful ITS in commercial and educational environments utilized by hundreds of thousands of students every year [136, 137]. It is reported that CT for mathematics are now in use in more than 2,500 schools across the US for 500,000 students per year [4].

It provides support for guided learning by doing [138], allocates students problems individually, monitors solution stages for students, offers context-sensitive feedback and hints, and applies a mastery study criteria [139]. A number of studies have shown that CT help raise students' mathematics achievement relative to traditional mathematics courses [139].

CT was built upon the assumptions of Adaptive Control of Thought–Rational (ACT-R) model theory of cognition and learning and was equipped with a built-in cognitive model [138]. Figure 3.1 displays the interface of CT for Algebra I. CT presents a problem description and asks several questions, students answer the questions by

[4] see https://pslcdatashop.web.cmu.edu/KDDCup/rules.jsp#the-challenge



- **Row**: the row number, 1...n for the training file and 1...n for the test file.
- **Anon Student Id**: unique, anonymous identifier for a student
- **Problem Hierarchy**: the hierarchy of curriculum levels containing the problem.
- **Problem Name**: unique identifier for a problem
- **Problem View**: the total number of times the student encountered the problem so far.
- **Step Name**: each problem consists of one or more steps, the step name is unique within each problem, but there may be collisions between different problems, so the only unique identifier for a step is the pair of problem_name and step_name.
- **Step Start Time**: the starting time of the step. Can be null.
- **First Transaction Time**: the time of the first transaction toward the step.
- **Correct Transaction Time**: the time of the correct attempt toward the step, if there was one.
- **Step End Time**: the time of the last transaction toward the step.
- **Step Duration (sec)**: the elapsed time of the step in seconds, calculated by adding all of the durations for transactions that were attributed to the step. Can be null (if step start time is null).
- **Correct Step Duration (sec)**: the step duration if the first attempt for the step was correct.
- **Error Step Duration (sec)**: the step duration if the first attempt for the step was an error (incorrect attempt or hint request).
- **Correct First Attempt**: the tutor's evaluation of the student's first attempt on the step—1 if correct, 0 if an error.
- **Incorrects**: total number of incorrect attempts by the student on the step.
- **Hints**: total number of hints requested by the student for the step.
- **Corrects**: total correct attempts by the student for the step. (Only increases if the step is encountered more than once.)
- **KC(KC Model Name)**: the identified skills that are used in a problem, where available. A step can have multiple KCs assigned to it. Multiple KCs for a step are separated by ~~ (two tildes).
- **Opportunity(KC Model Name)**: a count that increases by one each time the student encounters a step with the listed knowledge component. Steps with multiple KCs will have multiple opportunity numbers separated by ~~.

Figure 3.2: The attributes contained in each interaction record.

filling in the worksheet [3]. It checks and records every action performed by students, and displays the error messages just-in-time in the hint window if the student make a mistake. It also calculates the learned skills by using knowledge tracing and presenting them on a bar chart called Skillometer.

CT provides many subjects for students to learn, for example, the Algebra I and the “Bridge to Algebra” program for pre-Algebra. Students’ learning logs are collected during their interaction with the system. The attributes contained in each interaction record is shown in Figure 3.2. Important information about the learner and the learning process is recorded, such as the learner ID, the problem name, duration time, and correctness. Three development datasets collected by CT have been published in the KDD Cup 2010 Education Data Mining Challenge hosted by the PLSC Datashop [5]. In this thesis, we main use two of the datasets: Algebra I 2005-2006 and Bridge to Algebra 2006-2007.

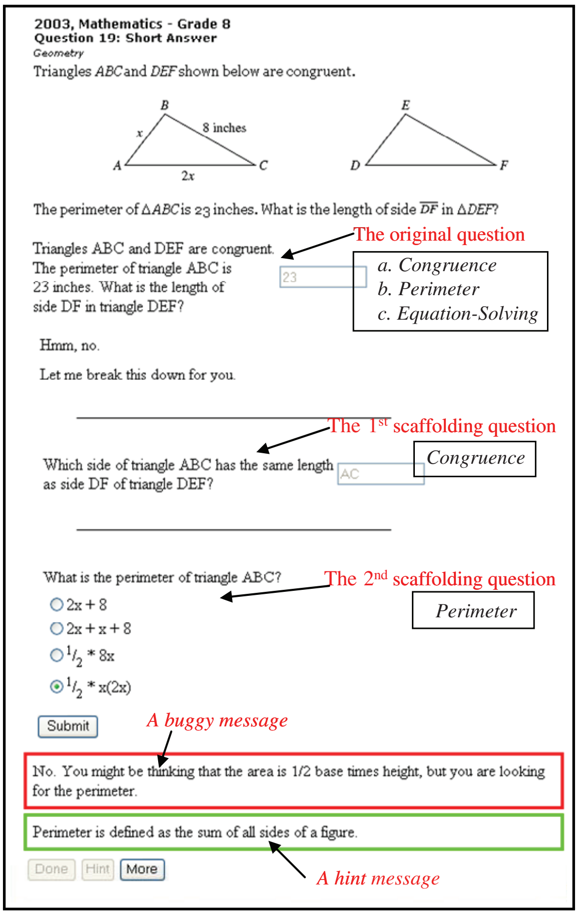
[5] https://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

3.1.2 ASSISTment System

ASSISTments is a free, online, formative assessment math platform utilized nationally in grades 3-10 by more than 18,000 teachers and 500,000 students. It was first created in 2004 and hosted by Worcester Polytechnic Institute. Assistance and assessment are incorporated together into the system that teaches students while giving them a more thorough assessment of their knowledge [4]. It helps students to solve difficult questions by dividing them into sub-steps, while it gathers data on their learning performance. After analyzing the rich data, numerous reports have been produced about the individual students to enable educators and stakeholders better understand the achievements and growth of students [140]. Moreover, different from other ITSs, the ASSISTments brings teachers, students, and researchers together as part of the ASSISTments ecosystem [140]. it is not only a test preparation program, but also a tool for building tutors. The authoring tool allows teachers to write individual ASSISTments (composed of questions with answers and associated hints, solutions, web-based videos, etc.) or to use pre-built ASSISTments, bundle them together in a problem set, and assign these to students [140].

Figure 3.3 shows the process of a student working on a problem in ASSISTments. The system first provides students an original problem, if students get the problem correct they are given a new one. If they get it wrong, they are provided with a tutoring session where a few scaffolding questions that break the problem down into steps will be offered [4]. Buggy and hint message can be also presented under the learning context. The student interaction process is recorded by the system as logs, in which more than 30 attributes are allocated for a piece of record [6]. The owner of the system has published three datasets collected in different years, including the ASSISTments 2009-2010, 2012-2013, and 2015 Skill Builder Dataset. These datasets are popularly used in the education data mining community. In this thesis, we mainly use two of the datasets: ASSISTments 2009-2010 and ASSISTments 2012-2013.

[6] The interpretation of the attributes for learning records can be found here: https://sites.google.com/site/assistmentsdata/how-to-interpret



2003, Mathematics - Grade 8
Question 19: Short Answer
Geometry

Triangles *ABC* and *DEF* shown below are congruent.

The perimeter of Δ*ABC* is 23 inches. What is the length of side $\overline{DF}$ in Δ*DEF*?

Triangles ABC and DEF are congruent. The perimeter of triangle ABC is 23 inches. What is the length of side DF in triangle DEF?

23

Hmm, no.

Let me break this down for you.

Which side of triangle ABC has the same length as side DF of triangle DEF?

AC

What is the perimeter of triangle ABC?

○ 2x + 8
○ 2x + x + 8
○ $^1/_2$ * 8x
● $^1/_2$ * x(2x)

No. You might be thinking that the area is 1/2 base times height, but you are looking for the perimeter.

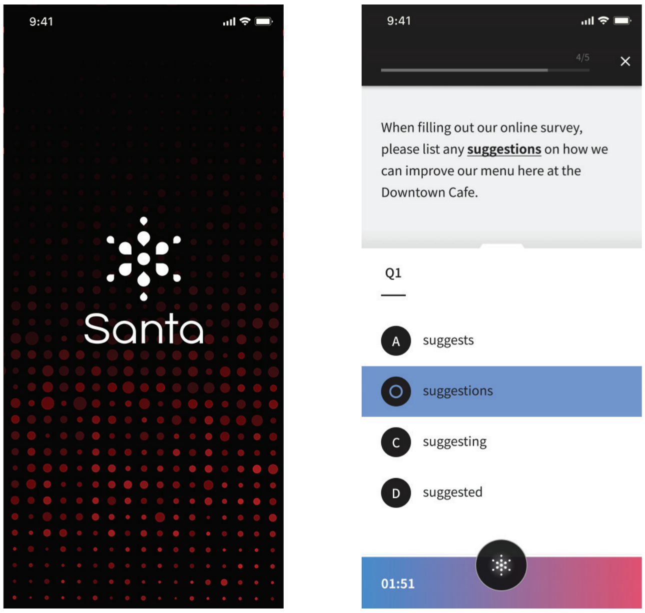
Perimeter is defined as the sum of all sides of a figure.

Figure 3.3: The ASSISTment system showing a student working on a problem [4].

3.1.3 Riiid's Santa TOEIC

Santa is a multi-platform, self-study solution equipped with artificial intelligence tutoring system developed by Korean Riiid Inc. It aids students in preparing for the TOEIC (Test of English for International Communication) listening and reading test [141]. The user interface is shown in Figure 3.4. Santa currently has 1,047,747 registered users and is available on both Android and iOS [5].

It uses AI techniques to provide not only the most appropriate level of contents but also the precise score prediction. Moreover, it motivates learners by visualizing their current learning status in contrast to their own target scores as well as comparison to average user score trends. Users receive personalized analysis report including the TOEIC prediction score, the five-point index consisting of listening, reading,



When filling out our online survey, please list any **suggestions** on how we can improve our menu here at the Downtown Cafe.

Figure 3.4: The user interface of Santa app [5].

vocabulary, grammar, and structure. It is reported that this app motivates the users to solve 3 times as more questions than they normally would with a workbook [7].

Based on this app, a new dataset named EdNet is published, which is the largest public available dataset in education field in terms of the total number of students, interactions, and interaction types [141]. There are 4 versions of EdNet datasets (EdNet-KT1,..., EdNet-KT4) that recording user behaviour ranging from basic exercising activity to complete interaction actions with other types of learning materials at increasing levels of detail. In this thesis, we only use the EdNet-KT1 dataset with pure exercising logs.

3.2 Datasets

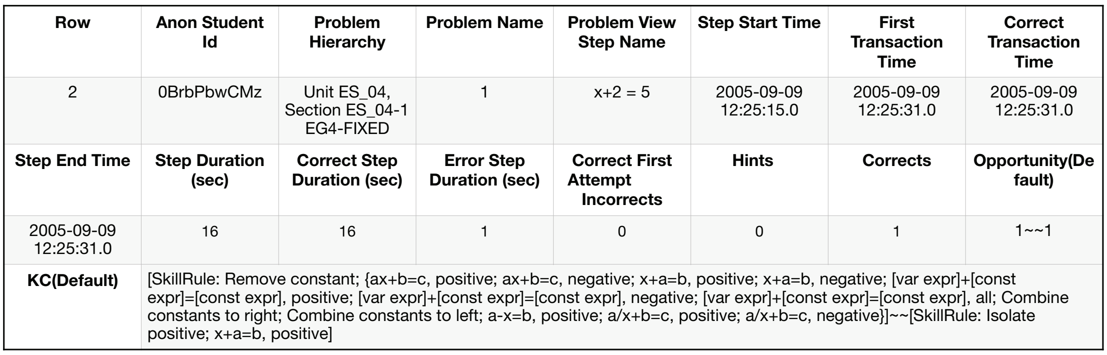
This section introduces the six datasets used in the following sections of this thesis. These datasets are all well-established and popular temporal datasets taking the form of interaction records between students and the real ITSs. They incorporate rich information regarding students and items as well as their interactions over time; hence,

[7] See https://www.riiid.co/

they are very suitable for dynamically assessing students' changing knowledge over a long time in large-scale scenarios.

3.2.1 Algebra0506

Algebra0506[8] was collected during 2005 and 2006 using Carnegie Learning's Cognitive Tutor. It contains the interaction logs of 569 learners and 173,113 items in the algebra field, resulting in 607,000 entries, and has the maximum number of items. Each item in this dataset contains one or more underlying skills. The average skill per item is 1.363. Each learner can attempt one item more than once, resulting in different attempts. The exercising history is recorded in chronological order. During the preprocessing, we concatenate the problem and step ID as a new problem ID, which is recommended by the challenge organizers, because the problems are typically divided into several steps. Figure 3.5 shows a piece of example record collected in this dataset.



Row	**Anon Student Id**	**Problem Hierarchy**	**Problem Name**	**Problem View Step Name**	**Step Start Time**	**First Transaction Time**	**Correct Transaction Time**
2	0BrbPbwCMz	Unit ES_04, Section ES_04-1 EG4-FIXED	1	x+2 = 5	2005-09-09 12:25:15.0	2005-09-09 12:25:31.0	2005-09-09 12:25:31.0
Step End Time	**Step Duration (sec)**	**Correct Step Duration (sec)**	**Error Step Duration (sec)**	**Correct First Attempt Incorrects**	**Hints**	**Corrects**	**Opportunity(Default)**
2005-09-09 12:25:31.0	16	16	1	0	0	1	1~~1
KC(Default)	[SkillRule: Remove constant; {ax+b=c, positive; ax+b=c, negative; x+a=b, positive; x+a=b, negative; [var expr]+[const expr]=[const expr], positive; [var expr]+[const expr]=[const expr], negative; [var expr]+[const expr]=[const expr], all; Combine constants to right; Combine constants to left; a-x=b, positive; a/x+b=c, positive; a/x+b=c, negative}]~~[SkillRule: Isolate positive; x+a=b, positive]						

Figure 3.5: A piece of example data collected from Algebra0506.

3.2.2 Bridge2Algebra0607

Bridge2Algebra0607[9] was collected in Carnegie Learning's Cognitive Tutor in 2006 and 2007, has been widely used in many papers after the release of the KDD Cup 2010 EDM Challenge. It contains the exercising records of 1130 learners attempting 129,263 math problems, with 1.8 million interaction entries. It is also a dataset with multi-skill items.

[8] Algebra0506:http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

[9] Bridge2Algebra0607:http://pslcdatashop.web.cmu.edu/KDDCup/downloads.jsp

The average skill per item is 1.013, indicating that most of the items still have only one skill. It has the similar record structure with the Algebra0506.

3.2.3 Assist0910

Assist0910[10] was collected from 2009 to 2010 using the ASSISTment system. The system is web-based and hence accessible anywhere/anytime [4], a large number of students have interacted with the system and their log data has been recorded. This dataset contains the interactions of 3002 learners and 17,705 items associated with 123 skills, forming 277,540 interaction records. A data quality issue has been detected in this dataset, in this thesis we use the corrected version of the skill-built dataset.

3.2.4 Assist1213

Assist1213[11], which updates Assist0910, was collected over the whole year of 2012. This dataset contains the columns related to affect that is tentatively utilized for affect prediction from the interaction data. However, in this thesis we do not use the affect column as the main objective of it is to assess learner knowledge. In this dataset, 22,591 learners in the system attempted 52,855 mathematics questions requiring 265 skills. This dataset contains the largest number of learners, questions, and interaction entries (nearly 2.7 million); however, the average attempted items per user are very low (118.73). The whole correctness for all items is also very low, at 0.6959. It is a one-skill-per-question dataset, meaning that each question requires one skill.

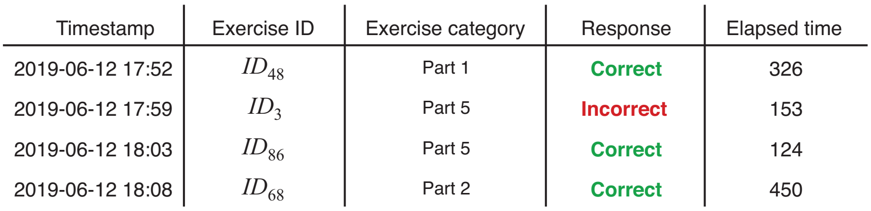
3.2.5 EdNet

EdNet[12] is newly available to the public. It is collected from the Santa, a multi-platform ITS available on iOS, Android and Web [141]. EdNet contains data collected from both mobile and desktop users in a consistent manner. The dataset consists of multiple-choice exercises in the TOEIC level with corresponding learner responses. Four versions of EdNet datasets (EdNet-KT1,..., EdNet-KT4) are released at increasing levels of detail. In

[10] Assist0910:https://sites.google.com/site/assistmentsdata/home/assistment-2009-2010-data/skill-builder-data-2009-2010

[11] Assist1213:https://sites.google.com/site/assistmentsdata/home/2012-13-school-data-with-affect

[12] EdNet:http://bit.ly/ednet_kt1



Timestamp	Exercise ID	Exercise category	Response	Elapsed time
2019-06-12 17:52	ID_{48}	Part 1	Correct	326
2019-06-12 17:59	ID_{3}	Part 5	Incorrect	153
2019-06-12 18:03	ID_{86}	Part 5	Correct	124
2019-06-12 18:08	ID_{68}	Part 2	Correct	450

Figure 3.6: A learner's interaction records on four problems in EdNet [5].

this thesis, we only use the EdNet-KT1 dataset with pure exercising logs from January 1st, 2019 to June 1st, 2020. Following existing work [41], we randomly selected 5000 students who answered 12,372 questions requiring 188 skills, thus obtaining 347,866 interaction logs. This dataset contains the minimum number of attempted items per learner (69.57) but the largest number of skills per item (2.28) among all the datasets. A learner's interaction records on four problems is shown in Figure 3.6. Besides, a file containing the mapping of questions and skills is also given.

3.2.6 Statics2011

Statics2011[13] contains the exercising records of 332 students attempting 1223 items from an engineering statics course at Carnegie Mellon University for 4 months (i.e., August to December) in 2011. Each item contains only one skill. This dataset has the smallest number (189,292) of interaction entries because it has the smallest number of students. It is published in the Datashop upon request.

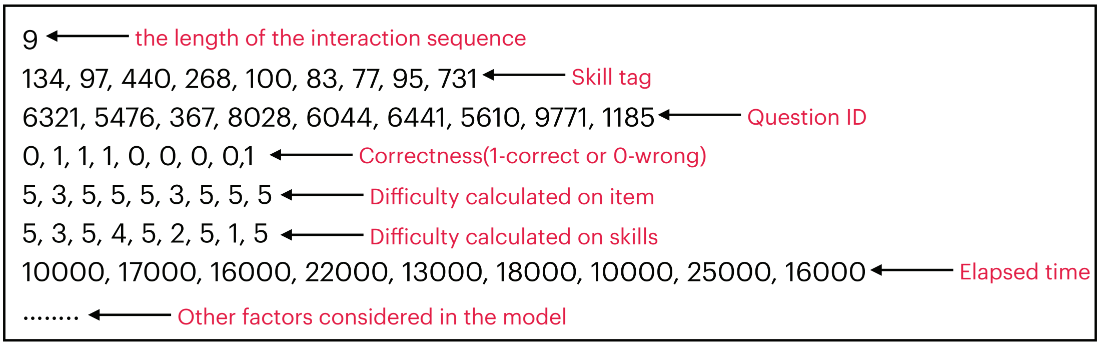
3.2.7 Preprocessing and Formatting

Preprocessing was conducted on all of the datasets. For the Algebra0506 and Bridge2Algebra0607 datasets, problems are typically divided into several steps; hence, we concatenated the problem and step ID as a new problem ID, which is recommended by the KDD Cup Challenge organizers. For other datasets, we used the *problem_id* as the item ID. To avoid noise, following the existing studies [30, 41, 29], we delete users with fewer than 10 interaction entries and questions with "not-a number" (NaN) skills from all the datasets. Table 3.1 summarizes the statistics of the six datasets.

[13] Statics2011:https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507

Table 3.1: Statistics of the six datasets used in this thesis.

Dataset	Algebra0506	Statics2011	Assist0910	Bridge2Algebra0607	Assist1213	EdNet
# of learners	569	332	3002	1130	22,591	5000
# of items	173,113	1223	17,705	129,263	52,855	12,372
# of skills	112	156	123	493	265	188
# of Interactions	607,000	189,292	277,540	1,817,360	2,682,439	347,864
# skill per item	1.363	1	1.20	1.013	1	2.28
# of attempted items per learner	1066.80	570.16	92.45	1608.28	118.73	69.57
Correctness	0.7553	0.7654	0.6585	0.8321	0.6959	0.5975
Sparseness	0.62%	46.62%	0.52%	1.24%	0.22%	0.56%
collecting period	1 year	4 months	1 year	1 year	1 year	1.5 years



```
9
134, 97, 440, 268, 100, 83, 77, 95, 731
6321, 5476, 367, 8028, 6044, 6441, 5610, 9771, 1185
0, 1, 1, 1, 0, 0, 0, 0,1
5, 3, 5, 5, 5, 3, 5, 5, 5
5, 3, 5, 4, 5, 2, 5, 1, 5
10000, 17000, 16000, 22000, 13000, 18000, 10000, 25000, 16000
........
```

the length of the interaction sequence
Skill tag
Question ID
Correctness(1-correct or 0-wrong)
Difficulty calculated on item
Difficulty calculated on skills
Elapsed time
Other factors considered in the model

Figure 3.7: The multiple line format for storing the interaction sequence records of a learner.

Note that we measure the sparseness of each dataset using "# of attempted items per learner" divides "# of items". As shown in the table, all the datasets except Statics2011 have quite low sparseness.

For the convenience of inputting data into the model, we join the multiple skills in an item into a new skill tag, similar to that conducted in a previous study [10, 142] and keep the mapping of new skill tag and the previous multiple skill tag. The processed datasets are stored in a uniform format (multiple line format) to represent the interaction sequence records, as shown in Figure 3.7, which can be found in [10]. In this format, multiple lines are composed of an interaction sequence. The first line indicates the length of the interaction sequence, and the second and third lines represent the skill tag and exercise id. The fourth line stands for correct answer (i.e., 1) or wrong answer (i.e., 0). The fifth and sixth lines are the item difficulties calculated in different methods, as will described in the Chapter 4. The seventh line records the elapsed time in milliseconds of a learner on each exercise. Other factors involved in the models can also be added in this data format.

After preprocessing and data formatting, each dataset is transformed into a file in the above multiple line format, as well as some additional files recording information, such as the q-matrix, the mapping of new and previous skill tags and the name of skills.

3.3 Evaluation Metrics

As described in Section 1.2.2, we assess learners' knowledge from their exercising logs, the smaller the gap between the estimated and learners' real knowledge states, the more accuracy of the built models. However, there is no ground-truth of the learners' real knowledge, the obtained assessment results cannot be directly evaluated. Researches in the existing studies have verified that the effectiveness of the estimated learner knowledge can be validated by predicting learner scores on the exercises. Hence in the practical implementation, model performance is evaluated by comparing the predicted responses and the real responses of learners on the exercises.

Three metrics are widely used on this task for measuring the performances of different models: prediction accuracy (ACC), area under the curve (AUC) and negative log-likelihood (NLL).

ACC:

$$ACC = \frac{TP + TN}{N} \tag{3.1}$$

where TP = True positive, TN = True negative, and N is the total number of samples.

AUC: stands for "Area under the ROC Curve", which measures the entire two-dimensional area underneath the entire ROC curve. AUC is a better measure than accuracy [143] and is widely used in the comparison of model performance. It considers the probability (be it 0.51 or 0.99) of the prediction results, and provides an overall performance measurement over all possible classification thresholds. AUC is interpreted as the probability that the model will score a random positive example more highly than a random negative one [14].

AUC score is between 0 and 1. A model with 100% false predictions has an AUC of 0.0, and AUC of 1.0 represents 100% correct predictions. Generally, the value 0.5 of AUC or ACC indicates the result by randomly guessing, the larger, the better.

NLL:

$$\text{NLL} \; = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(p_i\right) \tag{3.2}$$

where y_i is the real result, p_i is the perdicted result, and N is the number of samples. For NLL, the smaller the value, the better the model performance.

In the following chapters, we will show the performance of the three proposed models and compare them with the existing models using these metrics.

[14] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

4

Learner Knowledge Assessment by Modeling the Dynamic Knowledge Construction Procedure and Cognitive Item Difficulty

Human knowledge acquisition is an extremely complicated procedure, as we dynamically learn and forget over time. Cognitive psychology has long verified that the knowledge acquisition procedure is not only related to the learners but also the learning materials in the domain. To model this complex procedure, many factors need to be considered to make the model accurately assess learners' real knowledge. This is the **Issue 1** investigated in this thesis.

The material in this chapter is based on [12, 144]

In this chapter, we empirically examine the factors (learner factors: learning and forgetting, item factor: item difficulty) that influencing the learning performance, propose methods to quantify these factors and utilize them into our proposed model, named KTM-DLF, to assess the learner knowledge. Section 4.1 introduces the motivation of our proposed solution in this chapter; Section 4.2 gives an overview of our proposed framework; the detailed description of the proposed KTM-DLF model is presented in Section 4.3; Section 4.4 introduces the experimental settings and Section 4.5 presents the results of performance evaluation of the proposed model; a summary of this chapter is given in Section 4.6.

4.1 Motivation

Knowledge Construction Process Cognitive diagnostic models (e.g., Deterministic Inputs, Noisy "And" gate, DINA [62]) and data mining techniques (e.g., Boolean matrix factorization [66]) have been widely used in characterizing students' implicit knowledge proficiency from a static perspective. However, students' knowledge construction process is not static but evolves over time [28] since students learn and forget over time, making tracing students' knowledge inherently difficult. These learning and forgetting procedures have long been verified by two classical theories in educational psychology: the learning curve theory [54] and the Ebbinghaus forgetting curve theory [55]. The former uses learning curves to represent how an increase in learning performance comes from greater trials or exercises. The latter argues that students' knowledge states or levels can exponentially decline over time (i.e., memory decay).

Several studies on learner knowledge assessment have been conducted (partially) considering these two dynamic procedures [30, 120, 29, 28, 63, 96, 145], showing some benefit from adding temporal information for this task. However, some issues still remain. Most of the models consider only a fragment of the information related to learning or forgetting. Deep learning approach (e.g., deep knowledge tracing [10]) models students' learning by considering students' performance on a sequence of exercises over time, while the forgetting factor is not explicitly considered, and it can only be adopted to items with single KC. Factor analysis approach (e.g., knowledge tracing machine [30]) incorporates both the number of trials and the temporal information from the previous interactions to model learning and forgetting procedures. However, these factor analysis models use features obtained from the interactions either at the skill or the item level, or just neglect that learning and forgetting factors are closely related to the exercises done at each time. When fitting student models, it is better to rely on the totality of information available at hand, as verified by Vie in [30].

Cognitive Item Difficulty Problem difficulty undoubtedly has some influence on student performance [93, 50, 96, 115]. Different KCs involve different levels of difficulty and different problems with different combinations of various KCs can also exhibit different levels of difficulty. The relative difficulty level of a specific problem varies from student to student. For a specific student, the difficulty level of the same problem also varies over time in his cognition. However, existing KT work either does not consider problem difficulty or assumes it remains constant [50, 96, 115, 10], and this is unrealistic in the actual learning process. These studies assume that it is only the KCs involved that contribute to the difficulty of an item, hence given an item with the definite set of KCs associated with the item, the difficulty of that item is also fixed and will not change for different students over time, i.e. the item difficulty is item-oriented. However, researches [146, 147, 148, 149] from cognitive psychologists show that while KCs are of major relevance in problem solving, it is only one of several sources of difficulty, there are also some residuals that cannot be accounted for by the KCs, for example, the item types (multiple-choice question or short answer question), the memory loads imposed by different problem isomorphs (item structure) as well as the search space for finding the correct combinations of KCs to solve the problem. Moreover, in human learning, the same item is generally on different levels of difficulty in terms of the cognitive challenge it presents to different learners, hence considering the cognitive difficulty of items for different learners (i.e. user-oriented difficulty) will make the KT more accuracy for specific individuals.

To solve the challenges mentioned above, we herein propose a novel model, KTM-DLF (Knowledge Tracing Machine by modeling cognitive item Difficulty and Learning and Forgetting), to trace the evolution of each learner's knowledge acquisition during exercise activities by modeling his or her dynamic knowledge construction procedure and cognitive item difficulty. Specifically, we first specify the concept of cognitive item difficulty and propose a method to model the cognitive item difficulty adaptively based on learners' learning histories. Then, based on two classical theories (the learning curve theory and the Ebbinghaus forgetting curve theory), we propose methods for modeling learners' learning and forgetting over time. Finally, the KTM-DLF model is proposed to incorporate learners' abilities, the cognitive item difficulty, and the two dynamic procedures (learning and forgetting) together. Inspired by [30, 29], We then use the factorization machine framework to embed features in high dimensions and model pairwise interactions to increase the model's accuracy. Extensive experiments have been conducted on three public real-world datasets, and the results confirm that our proposed model outperforms the other state-of-the-art educational data mining models.

The contributions of this proposed method can be summarized as follows:

- We propose a novel knowledge tracing model named KTM-DLF that traces the evolution of students' knowledge acquisition over time by explicitly modeling students' learning and forgetting behaviors as well as the cognitive item difficulty.
- We propose a concept of cognitive item difficulty and a method to model this user-oriented difficulty adaptively in terms of the cognitive challenge it presents to different individuals.
- We model students' learning and forgetting behaviors by taking account of their memory decay and the benefits of attempts when an item can involve multiple KCs.
- Experiments on real-world public datasets shows the effectiveness of the KTM-DLF model compared with the state-of-the-art models.

4.2 Solution Overview

In this section, we give an overview the proposed KTM-DLF model for KT.

As introduced in Section 1.2.2, KT is a supervised learning problem: given the labelled past exercise log, it predicts the future performance. Figure 4.1 shows the framework of the proposed KTM-DLF model. It mainly consists of two stages: modeling and predicting.

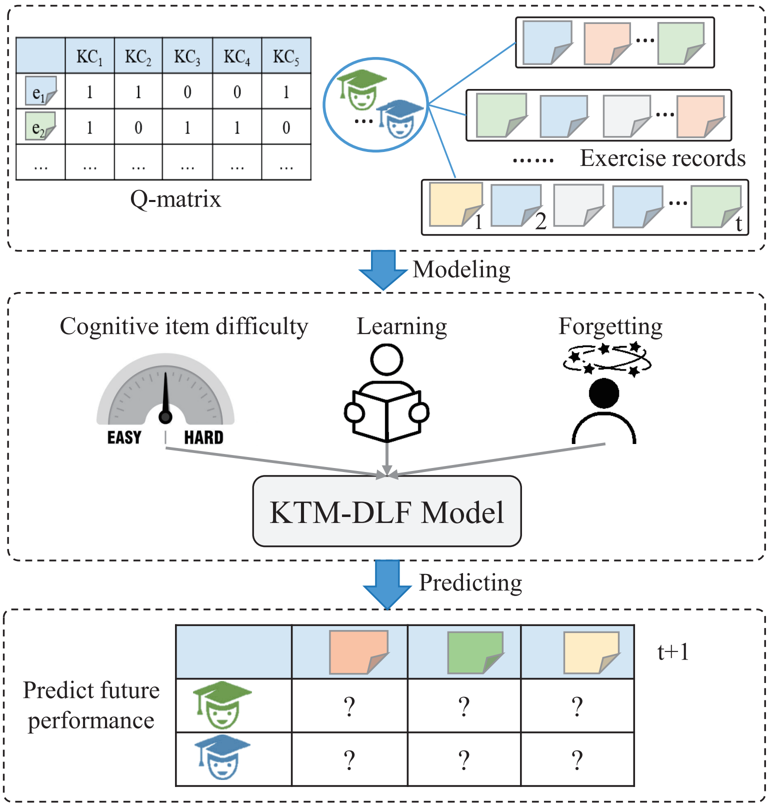


Figure 4.1: The process of knowledge tracing based on the proposed KTM-DLF model.

In the first stage, we propose the concept of cognitive item difficulty, which is different from the general item difficulty. General item difficulty is constant based on the involved KCs to all the students, and it is item-oriented, while the cognitive item difficulty not only consider the inherent item difficulty as the general item difficulty, but also consider the cognitive factors of users, which makes it user-oriented. This is consistent with human learning experience as the same item is generally on different levels of difficulty in terms of the cognitive challenge it presents to different respondents, hence considering the cognitive item difficulty for different individuals will make the KT more accuracy for specific individuals. We propose a method to model the cognitive item difficulty. Moreover, as human knowledge construction procedure is dynamic due to learning and forgetting all the time, we also propose methods to model the learning and forgetting procedures based on their interaction history. Then by incorporating these factors together, we propose the KTM-DLF model, which models students' knowledge states by considering their ability, their learning and forgetting procedures, and the cognitive difficulty level of items.

In the second stage, we use the proposed KTM-DLF model to predict students' performance in the future interactions. In the following section, we will specify the modeling procedure in the KTM-DLF model.

4.3 Proposed KTM-DLF Model for KT

In this section, we introduce the proposed KTM-DLF model. In our setting, students learn to obtain a set of KCs by interacting sequentially with the tutoring systems. Their knowledge levels on KCs at specific times are explicitly measured by the ability to answer items involving the set of KCs. Moreover, students' knowledge proficiency can be enhanced by learning and can also decline over time as a result of forgetting.

Based on these assumptions, this section presents the proposed KTM-DLF model, which models students' knowledge states by considering their ability, their learning and forgetting procedures, and the cognitive difficulty level of items.

4.3.1 Modeling Cognitive Item Difficulty

Rationality of Cognitive Item Difficulty Several studies have already attempted to incorporate the item difficulty in KT, and the experimental results showed empirically the benefits of adding this difficulty information for this task [93, 16, 134, 30, 135]. However, nearly all of the existing models that formulate the difficulty at the item or skill level consider the difficulty coefficient as a constant, assuming that it will not change for different students over time. They assume that it is only the KCs involved that contribute to the difficulty of an item, hence given an item with the definite set of KCs associated with the item, the difficulty of that item is also fixed. This is unrealistic in the actual learning process as problem difficulty affects performance undoubtedly and also varies overtime in terms of the cognitive challenge it presents to individual learners.

Actually item difficulty is a subjective variable for learners and has been widely studied in the field of cognitive psychology. Some prior work [146, 147, 148, 149] evaluated the possible factors contributed to item difficulty. Kubinger et al. [147]

performed case study and showed some attributes lead to item difficulty, such as item types, item structures and knowledge depth. Kotovsky et al [149] demonstrated that difficulty was also correlated with the size of the memory loads imposed by the different problem isomorphs. In [146], Kotovsky et al proposed that while KCs are of major relevance in problem solving, it is only one of several sources of difficulty, another source of difficulty widely recognized is the size of the space that must be searched to find the correct path from start to goal from among the many paths available.

Based on these findings in the cognitive psychology and the limitations in the existing KT models, we proposed in this thesis the concept of "*Cognitive Item Difficulty*", which models the item difficulty based not only on the KCs involved in the item but also on other residual aspects (e.g. the memory loads of problem isomorphs and size of search space). Specifically, we model the cognitive item difficulty by considering the difficulty of KCs and the item itself, as well as the search space for each KC and item.

Quantification of Cognitive Item Difficulty Based on the above analysis, item difficulty can be expected to be determined not only based on the difficulty of each KC involved (a function of KCs), but also according to the characteristic of items and students' current knowledge states (their own cognition). Therefore, we quantify Cognitive Item Difficulty by taking into consideration not only the difficulty from the items and KCs themselves, but also from the cognitive aspects of each student. We call the former as the inherent difficulty of item or KC (part 1 and 2 in Eq. 4.1) as it is item-oriented (KCs are also mapped to an item) and is also general to all the users. Notably, we model the residual difficulty that not accounted for by the involved KCs into a term associated with the item (part 1 in Eq. 4.1).

Moreover, different students have their own different knowledge structures at different times, the search space for answering an item (or KC) built during their previous learning experiences also varies, hence we try to model the search space from students' own cognition to make the item difficulty user-oriented, a reason why we call it cognitive item difficulty.

However, the search space for answering an item (or KC) is implicit and cannot be directly measured. Intuitively, the more difficult to search the correct path to answer an item in the search space, the more tendency a student will get an incorrect answer.

Since students' previous exercising records are explicit and accessible, we can use the incorrect interactions in students' previous exercising process as an indicator to measure each student's search space that contributing to the cognitive difficulty of items (part 3 in Eq. 4.1).

Given the above discussion, the cognitive difficulty $d(i, j, t)$ of item j for student i at time t is defined as

$$d(i, j, t) = \underbrace{\delta_j}_{Part-1} + \underbrace{\sum_{k \in KC(j)} \beta_k}_{Part-2} + \underbrace{\theta_m \Psi_{i,j,t} + \theta_n \left[\frac{\sum_{k \in KC(j)} \Psi_{i,k,t}}{|KC(j)|} \right]}_{Part-3} \tag{4.1}$$

where β_k is the inherent difficulty of skill k, and δ_j is a term denoting the inherent residual difficulty of item j that cannot be accounted for by the involved KCs. $KC(j)$ is the set of skills required to solve item j. The first two terms of Eq. 4.1 are item-oriented and they are general inherent difficulty for all the students. The last two terms of Eq. 4.1 are indicators that measuring the search space for a specific item and the involved KCs, respectively. They are adjustive terms to make the cognitive item difficulty user-oriented based on a student's current knowledge structure. Inspired by [93], the terms for measuring incorrect interactions on items $\Psi_{i,j,t}$ and on KCs $\Psi_{i,k,t}$ are defined in Eq. 4.2. θ_m and θ_n are the biases for the cognitive difficulty levels in the search space indicated by previous attempts on the same item and on the associated skills.

$$\Psi_{i,v,t|v=\{j,k\}} = \begin{cases} \left[\frac{\left|\{x_{i,v}==0\}\right|_{0:t}}{\left|N_{i,v}\right|_{0:t}} * (c-1) \right], & if \left|N_{i,v}\right|_{0:t} \geq 5 \\ c, & else \end{cases} \tag{4.2}$$

where $\Psi_{i,j,t}$ and $\Psi_{i,k,t}$ are quantified into $c + 1$ levels (ranging from zero to c). $N_{i,v}$ is the set of problems or skills the student i has attempted prior to time t, and $x_{i,v}$ is the outcome of the attempt by student i on problem j or skill k. An outcome of zero is a failure. If a student has attempted a problem or skill fewer than five times, the level will be set as c indicating the highest level of difficulty in the search space.

4.3.2 Modeling Student Learning and Forgetting

Existing KT approaches model students' learning in an implicit manner by obtaining their (implicit) knowledge states through learning from sequences of multiple attempts. There are only a few studies in the field that have addressed learning and forgetting explicitly and simultaneously [120, 131, 132, 29, 28, 133], while either simplifying the forgetting behavior or just ignoring it.

Actually, learning and forgetting are two widely accepted procedures in educational psychology that can influence learning outcomes. The more exercises a student does, the bigger gain of knowledge proficiency he or her will obtain. Moreover, the longer the lag time from the previous interaction, the greater probability the student will forget something. Based on these assumptions, we define learning as follows.

$$l(i,j,t) = \Phi_{i,j,t} + \sum_{k \in KC(j))} \Phi_{i,k,t} \tag{4.3}$$

$$\begin{aligned} \Phi_{i,v,t|v=\{j,k\}} = \sum_{tw=1}^{T} \{ & \theta_{v,3tw+1} log(1 + W_{i,v,tw}) \\ & + \theta_{v,3tw+2} log(1 + F_{i,v,tw}) \\ & - \theta_{v,3tw+3} log(1 + A_{i,v,tw}) \} \end{aligned} \tag{4.4}$$

where learning $l(i,j,t)$ is composed of the acquisition from attempting both the same items ($\Phi_{i,j,t}$) and also different items containing the same set of skills ($\Phi_{i,k,t}$). $W_{i,v,tw}$ and $F_{i,v,tw}$ denote the number of attempts that skill or item v have been correctly and incorrectly recalled among $A_{i,v,tw}$ attempts in time window tw by student i. $tw|_{0:T}$ is a set of expanding time windows inspired by [29], which are not disjoint but span increasing time intervals. The consideration of both successful and failed attempts corresponds to the fact that being correct or incorrect in some items or skills both contribute to the knowledge acquisition.

Early studies of forgetting revealed that the retention rate decreases exponentially as time passes [131, 28]. The longer interval of the interaction with some knowledge, the more likely the forgetting occurs. For this reason, we formulate forgetting behavior as

$$f(i,j,t) = \theta_{j,j} e^{\Delta_{j,j}} + \theta_{k,k} \sum_{k \in KC(j))} e^{\Delta_{k,k}} + \theta_{j,j-1} e^{\Delta_{j,j-1}} \tag{4.5}$$

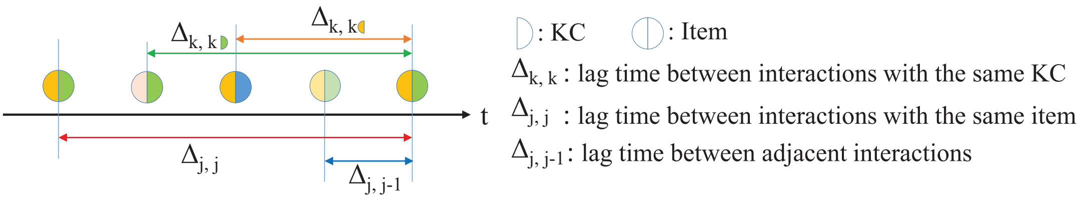

Δk, k
Δk, k
t
Δj, j
Δj, j-1
: KC
: Item
Δk, k : lag time between interactions with the same KC
Δj, j : lag time between interactions with the same item
Δj, j-1 : lag time between adjacent interactions

Figure 4.2: The three kinds of information related to forgetting from a student's sequence of interactions. Each semicircle represents a KC and each circle corresponds to an interaction with an item at a specific time and the same color represents the same KC.

where $f(i, j, t)$ can be interpreted as representing the amount of forgetting in a student's memory, which is composed of three parts, as shown in Figure 4.2, the lag time between the current interaction and the previous interaction with the same item $\Delta_{j,j}$, the lag time between the current interaction and the previous interaction with the same associated skill $\Delta_{k,k}$, and the lag time between adjacent interactions in the learning sequence $\Delta_{j,j-1}$. For some problems, these are related or similar, or the skills contained are related. Hence, the lag time between adjacent interactions in the sequence can affect the performances on these questions. Incorporating the time gap in the sequence into the model might capture this effect [120].

4.3.3 Proposed KTM-DLF Model

Based on the defined cognitive item difficulty and the modeling of learning and forgetting, this subsection proposes a KTM-DLF model for KT and leverage the factorization machine (FM) framework [150] to integrate these factors. This framework enriches the proposed model by embedding the features in high dimensions and modeling pairwise interactions between those features.

For an embedding dimension of $dim = 0$, our model is formulated as

$$\sigma(P(Y_{i,j,t} = 1)) = \alpha_{i,t} - d(i, j, t) + l(i, j, t) - f(i, j, t) \tag{4.6}$$

by incorporating Eq. 4.1, 4.3 and 4.5. Thus, the probability of student i correctly attempting item j at time t depends on the student's ability $\alpha_{i,t}$ at time t, the cognitive difficulty $d(i, j, t)$ of the item j to student i, and the student's learning $l(i, j, t)$ and forgetting $f(i, j, t)$ during this time period. $\sigma(.)$ here is a link function; in our implementation, we use *probit* as the link function. It is worth noting that when $dim = 0$, our model does not consider the interactions of the embedding features; it is actually a regression model. Our model is performed on a set of sparse vectors x of length N by encoding of all the features in the input samples.

For higher embedding dimensions $dim > 0$, all features are embedded in dim dimensions and their interactions are modeled in a pairwise manner, as shown in Figure 4.3. The quadratic term of our model is:

$$\phi_{KTM-DLF} = \sum_{1 \leq i < j \leq N} \langle v_i, v_j \rangle x_i x_j \tag{4.7}$$

where x_i and x_j are the i^{th} and j^{th} feature of the input sample, respectively. $v_j \in R^{dim}$ is the embedding vector of feature j for some dimension, and $\langle . \rangle$ is the inner product of two embedding vectors (as shown with the green arrow). As shown in Figure 4.3, all the features in the input sample are weighted in an element-wise matter, as the general regression analysis (as shown with the red arrow). Moreover, all the features are embed in high dimensions as the embedding vectors, and FM models the interaction by factorizing it, which allows high quality parameter estimates of higher-order interactions under sparsity [116]. These regression and pair-wise interaction procedures are both sent to the output units to predict students' performance, which can make our model fit student data in a more accuracy way.

Training of KTM-DLF is performed by minimizing the logistic loss over the observations and the outcomes.

$$logloss = \frac{1}{m} \sum_{i=1}^{m} log(1 + exp(-y_i(P_{X_i} = 1))) \tag{4.8}$$

Following previous work [29, 150], we also use a hierarchical distributional assumption to train our model. The regression and embedding weights for the feature vectors both follow a normal prior distribution $\mathcal{N}(\mu, 1/\lambda)$, and μ and λ hierarchically follow hyperpriors $\mu \sim \mathcal{N}(0, 1)$ and $\lambda \sim \Gamma(1, 1)$. Markov chain Monte Carlo Gibbs sampling is used to fit our model (see [150] for the details).

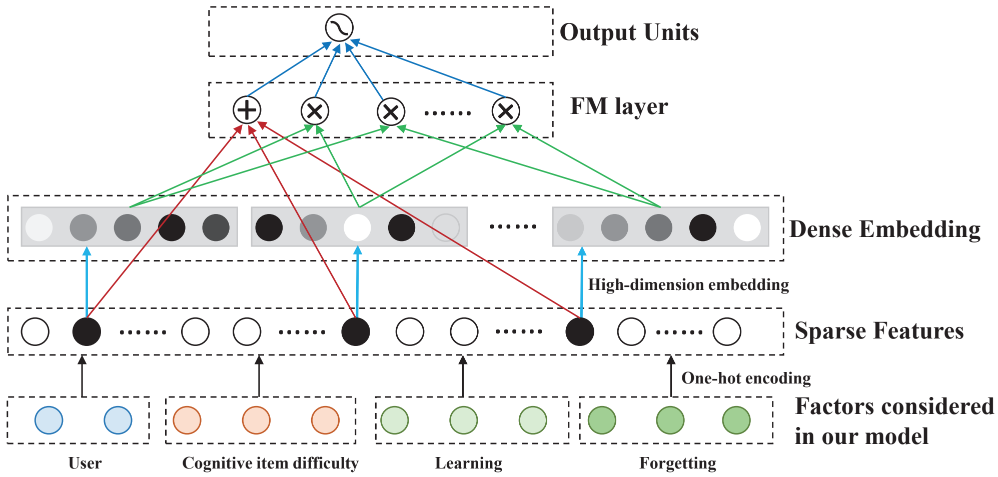


Figure 4.3: The framework of the KTM-DLF.

4.4 Experimental Setting

This section describes the experimental settings (datasets, implementation details and comparison baselines). The detailed experimental results and model analysis are presented in the next section.

4.4.1 Datasets

We used three public real-world datasets for our experiments: the Algebra 2005-2006 dataset, the Bridge to Algebra 2006-2007 and the ASSISTments 2012-2013 dataset. The details of these datasets are shown in Section 3.2.

4.4.2 Implementation Details

For each dataset, five-fold cross validation was performed at the student level. We divided all the students into five disjoint groups, and their interaction entries were separated into training and testing sets to perform the cross validation.

Our model was implemented in Python, *pywFM* [1] was used as a wrapper for $libfm$ [150] to implement the factorization machine for classifying data when the dimension of the model is greater than zero. In our experiments, our model was trained during 500

[1] https://github.com/jfloff/pywFM

epochs, because that was sufficient for convergence. For the adaptive item difficulty, we set the c in Eq. 4.2 as 5, indicating six levels of item difficulty (0 - 5). For the time windows used in Eq. 4.4, we used the same time windows as in [29]: {1 /24, 1, 7, 30, $+\infty$ } with time units expressed in days.

4.4.3 Comparison Baselines

We compared the proposed KTM-DLF model with six of the best known state-of-the-art KT models: DAS3H [29], DASH [132], IRT [50], MIRT [115], PFA [96], AFM [95]. These were chosen either because of their predominance in psychometrics or educational data mining, or because they are best performers. Moreover, these models also have explicit underlying theories and interpretability, as our proposed model does. Here we do not compare with the sequence model approach (e.g. DKT), as it can only be adopted to items with single KC, and there is also a mild controversy concerning the performance [29, 129, 130, 119]. Moreover, it is black-box to fit the data, thus making it very difficult to interpret their performance. Table 4.1 shows the comparisons between our proposed model and previous work. To the best of our knowledge, no KT model accounts for both students' dynamic knowledge construction procedure and cognitive item difficulty, a gap that we intend to bridge in this chapter.

We used the KTM framework [30] [2] and the code in [29] [3] to implement the baseline models and tested the models in a variety of dimensions ($d = 0, 5, 10, 20$). Note that we embed the features in high dimensions and model the regression and pair-wise interaction procedures by using the factorization machine framework, hence in all the baseline models, we took the factors considered in each model following the same above-mentioned procedures to test the models in various dimensions. ACC, AUC and NLL are used for measuring the performances of different models, the details of these metrics are shown in Section 3.3.

[2] https://github.com/jilljenn/ktm

[3] https://github.com/BenoitChoffin/das3h

Table 4.1: Comparisons between our proposed model and previous work. Note that "✓" ("-") indicates the factor is (not) considered, the "item" ("skill") in the learning columns indicates attempts considered at the item-level (skill-level), and the information in the column "Use of skill" indicates whether the model can be applied to problems with single or multiple KCs.

Method	User ability	Difficulty		Learning			Forgetting			Use of skill	Learn on ordered sequence
		Item level	Skill level	#Win	#Fail	#Attempt	Item time gap	Skill time gap	interaction time gap		
IRT [50]	✓	constant	-	-	-	-	-	-	-	single	-
AFM [95]	-	-	constant	-	-	skill	-	-	-	multiple	-
PFA [96]	-	-	constant	skill	skill	-	-	-	-	multiple	-
BKT [63]	✓	-	-	-	-	-	-	-	-	single	✓
DKT [10]	-	-	-	-	-	-	-	-	-	single	✓
Extended-DKT [120]	-	-	-	-	-	-	✓	-	✓	single	✓
DKT-DSC [121]	✓	-	-	-	-	-	-	-	-	single	✓
DKVMN [32]	✓	constant	-	-	-	-	-	-	-	single	✓
KTM [30]	✓	constant	constant	skill	skill	skill	-	-	-	multiple	✓
DASH [132]	✓	constant	-	item	-	item	-	✓	-	-	✓
DAS3H [29]	✓	constant	constant	skill	skill	skill	-	✓	-	multiple	✓
KTM-DLF	✓	adaptive	adaptive	Item, skill	Item, skill	Item, skill	✓	✓	✓	multiple	✓

Table 4.2: Comparisons among different models on the Algebra 2005-2006 dataset. ↑ (↓) indicates higher (lower) is better.

Model	dim	AUC ↑	ACC ↑	NLL ↓
KTM-DLF	5	**0.837** ± 0.003	**0.821** ± 0.006	**0.405** ± 0.007
KTM-DLF	0	**0.836** ± 0.002	**0.819** ± 0.009	**0.404** ± 0.013
KTM-DLF	10	**0.832** ± 0.003	**0.819** ± 0.008	**0.407** ± 0.011
KTM-DLF	20	0.826 ± 0.003	0.816 ± 0.004	0.415 ± 0.005
DAS3H	0	0.826 ± 0.003	0.815 ± 0.007	0.414 ± 0.011
DAS3H	5	0.818 ± 0.004	0.812 ± 0.007	0.421 ± 0.011
DAS3H	20	0.817 ± 0.005	0.811 ± 0.004	0.422 ± 0.007
DASH	5	0.775 ± 0.005	0.802 ± 0.01	0.458 ± 0.012
DASH	20	0.774 ± 0.005	0.803 ± 0.014	0.456 ± 0.017
DASH	0	0.773 ± 0.002	0.801 ± 0.004	0.454 ± 0.006
IRT	0	0.771 ± 0.007	0.800 ± 0.009	0.456 ± 0.015
MIRT	20	0.770 ± 0.007	0.800 ± 0.006	0.460 ± 0.007
MIRT	5	0.770 ± 0.004	0.800 ± 0.008	0.459 ± 0.011
PFA	0	0.744 ± 0.004	0.782 ± 0.003	0.481 ± 0.004
AFM	0	0.707 ± 0.005	0.774 ± 0.004	0.499 ± 0.006
PFA	20	0.670 ± 0.010	0.748 ± 0.005	1.008 ± 0.047
PFA	5	0.664 ± 0.010	0.735 ± 0.013	1.107 ± 0.079
AFM	20	0.644 ± 0.005	0.750 ± 0.005	0.817 ± 0.076
AFM	5	0.640 ± 0.007	0.742 ± 0.009	0.941 ± 0.056

4.5 Performance Evaluation

In this section, we evaluate the performance of our proposed model by comparing it with the state-of-the-art-models; ablation tests are also conducted to test different settings.

4.5.1 Performance Comparisons for Different Models

The comparisons among different models on three datasets are shown in Tables 4.2, 4.3 and 4.4. Five-fold cross-validation is performed in our experiments, and hence, the mean values of AUC, ACC and NLL are shown in these tables. Moreover, standard deviations over five folds are reported.

The results show that the proposed KTM-DLF model outperforms all of the other models on all three datasets. This makes sense since the KTM-DLF model takes into

Table 4.3: Comparisons among different models on the ASSISTments 2012-2013 dataset. ↑ (↓) indicates higher (lower) is better.

Model	dim	AUC ↑	ACC ↑	NLL ↓
KTM-DLF	5	**0.756** ± 0.001	**0.744** ± 0.002	**0.522** ± 0.003
KTM-DLF	10	**0.755** ± 0.002	**0.744** ± 0.001	**0.522** ± 0.002
KTM-DLF	20	**0.754** ± 0.002	**0.743** ± 0.002	**0.523** ± 0.002
KTM-DLF	0	0.745 ± 0.001	0.739 ± 0.002	0.540 ± 0.002
DAS3H	5	0.744 ± 0.002	0.737 ± 0.001	0.531 ± 0.001
DAS3H	20	0.740 ± 0.001	0.736 ± 0.002	0.533 ± 0.003
DAS3H	0	0.739 ± 0.001	0.736 ± 0.001	0.534 ± 0.002
DASH	0	0.703 ± 0.002	0.719 ± 0.003	0.557 ± 0.004
DASH	5	0.703 ± 0.001	0.720 ± 0.001	0.557 ± 0.001
DASH	20	0.703 ± 0.002	0.720 ± 0.002	0.557 ± 0.002
IRT	0	0.702 ± 0.001	0.719 ± 0.001	0.558 ± 0.001
MIRT	20	0.701 ± 0.001	0.720 ± 0.001	0.558 ± 0.001
MIRT	5	0.701 ± 0.002	0.719 ± 0.001	0.558 ± 0.001
PFA	5	0.669 ± 0.002	0.709 ± 0.002	0.577 ± 0.002
PFA	20	0.668 ± 0.002	0.709 ± 0.003	0.578 ± 0.003
PFA	0	0.668 ± 0.002	0.708 ± 0.001	0.579 ± 0.002
AFM	5	0.610 ± 0.001	0.699 ± 0.002	0.597 ± 0.001
AFM	20	0.609 ± 0.001	0.699 ± 0.003	0.597 ± 0.003
AFM	0	0.608 ± 0.002	0.697 ± 0.002	0.598 ± 0.002

consideration the cognitive item difficulty and the learning and forgetting procedure; thus, it incorporates more information regarding the knowledge acquisition process and hence can trace the changes in students' knowledge more accurately. AFM and PFA perform the worst on all of the datasets since they only incorporate the skill bias and numbers of attempts on the skills, and they consider the bias and weights for every skill as fixed. However, for the three datasets, the numbers of skills are far less than the numbers of items, and hence, it is a coarse way to model students' performances based on these limited fixed parameters. Given their simplicity, IRT and MIRT obtain better performance, which might be because they model student ability and item bias for every student and every item and this is much more suitable for the enormous datasets used in our experiments. By considering memory decay and skill bias, DAS3H obtains the second best performance and outperforms the DASH model on three datasets. Compared with DAS3H, our KTM-DLF model has +0.011 AUC improvement

Table 4.4: Comparisons among different models on the Bridge to Algebra 2006-2007 dataset. ↑ (↓) indicates higher (lower) is better.

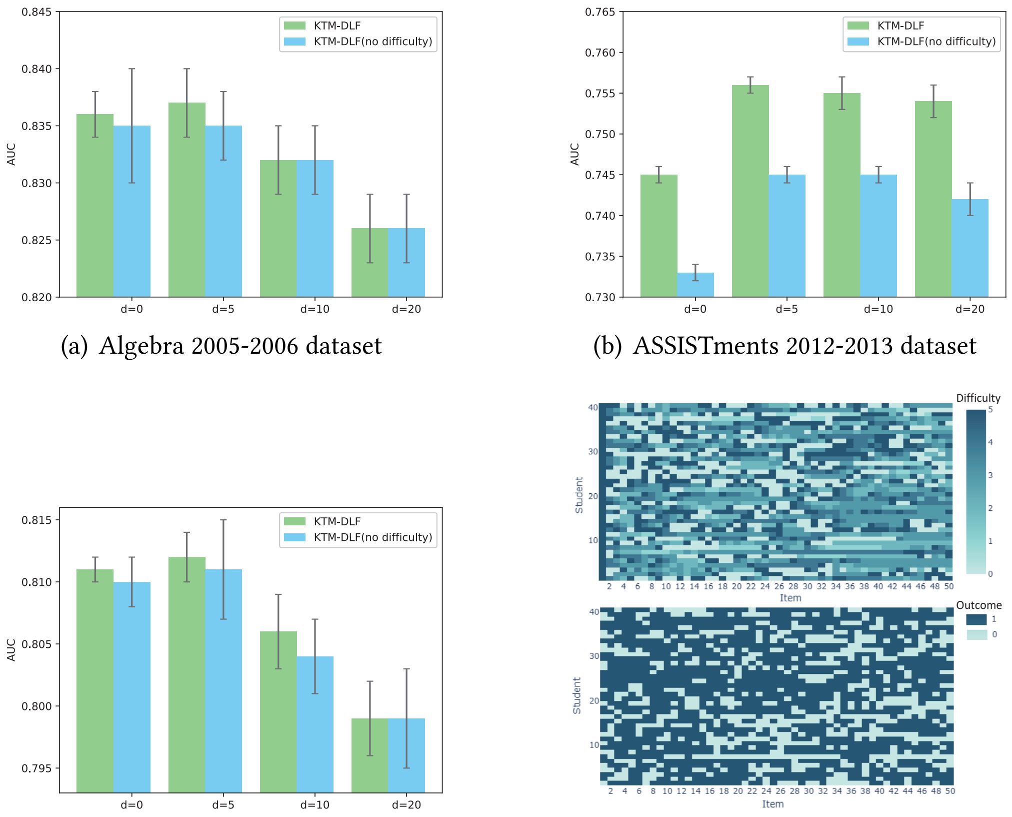
Model	dim	AUC ↑	ACC ↑	NLL ↓
KTM-DLF	5	**0.812** ± 0.002	**0.851** ± 0.003	**0.362** ± 0.005
KTM-DLF	0	**0.811** ± 0.001	**0.850** ± 0.003	**0.357** ± 0.005
KTM-DLF	10	**0.806** ± 0.003	**0.849** ± 0.006	**0.365** ± 0.010
KTM-DLF	20	0.799 ± 0.003	0.849 ± 0.003	0.370 ± 0.004
DAS3H	5	0.791 ± 0.005	0.848 ± 0.002	0.369 ± 0.005
DAS3H	0	0.790 ± 0.004	0.846 ± 0.002	0.371 ± 0.004
DAS3H	20	0.776 ± 0.023	0.838 ± 0.019	0.387 ± 0.027
DASH	0	0.749 ± 0.002	0.840 ± 0.005	0.393 ± 0.007
DASH	20	0.747 ± 0.003	0.840 ± 0.001	0.399 ± 0.002
IRT	0	0.747 ± 0.002	0.839 ± 0.004	0.393 ± 0.007
DASH	5	0.747 ± 0.003	0.840 ± 0.002	0.399 ± 0.002
MIRT	5	0.746 ± 0.002	0.840 ± 0.004	0.398 ± 0.006
MIRT	20	0.746 ± 0.004	0.839 ± 0.005	0.399 ± 0.007
PFA	20	0.746 ± 0.003	0.839 ± 0.002	0.397 ± 0.004
PFA	5	0.744 ± 0.007	0.838 ± 0.003	0.402 ± 0.007
PFA	0	0.739 ± 0.003	0.835 ± 0.005	0.406 ± 0.008
AFM	5	0.706 ± 0.002	0.836 ± 0.003	0.411 ± 0.004
AFM	20	0.706 ± 0.002	0.836 ± 0.003	0.412 ± 0.004
AFM	0	0.692 ± 0.002	0.833 ± 0.004	0.423 ± 0.006

on Algebra 2005-2006, +0.012 on ASSISTments 2012-2013 and +0.021 on Bridge to Algebra 2006-2007 dataset, which show its superiority in the performance of KT.

To further compare whether the performance of these KT models are statistically significant, we conducted two-tailed independent t-tests on the AUC results between our proposed KTM-DLF model and the other models, as shown in Table 4.5. Note that for each kind of model with different dimensions, we choose the model with dimension that obtaining best performance on AUC to compare with other models. When the variances of two groups of results are the same, student's t-test is used, otherwise Welch's t-test is used [151]. We found that the p-values between KTM-DLF and all the other models are all smaller than 0.005, which indicating that the null hypothesis is rejected and the differences between our model and the other models are statistically significant on all the three datasets.

Table 4.5: P-value comparisons of independent t-tests on AUC measure between KTM-DLF model and the other models on three datasets.

Dataset	p-value(KTM-DLF & the other models)					
	DAS3H	DASH	IRT	MIRT	PFA	AFM
Algebra 2005-2006	4.06E-04	1.33E-07	3.25E-06	2.99E-06	4.61E-10	1.07E-09
ASSISTments 2012-2013	2.34E-05	4.14E-09	3.95E-13	3.41E-13	2.25E-10	2.22E-16
Bridge to Algebra 2006-2007	2.57E-04	2.92E-11	2.28E-11	2.02E-11	1.45E-09	4.59E-13



(a) Algebra 2005-2006 dataset

(b) ASSISTments 2012-2013 dataset

(c) Bridge to Algebra 2006-2007 dataset

(d)

Figure 4.4: AUC comparisons of the effectiveness of cognitive item difficulty on three datasets. (a), (b) and (c) show the performance comparisons between KTM-DLF model and its ablated model without considering the cognitive aspects on Algebra 2005-2006, ASSISTments 2012-2013, and Bridge to Algebra 2006-2007 dataset, respectively. The upper part of (d) visualizes the cognitive item difficulty of the first 50 problems attempted by the first 40 students in the ASSISTments 2012-2013 dataset, and the lower part of (d) shows the real response outcome of the 50 problems attempted by the first 40 students in the corresponding dataset.

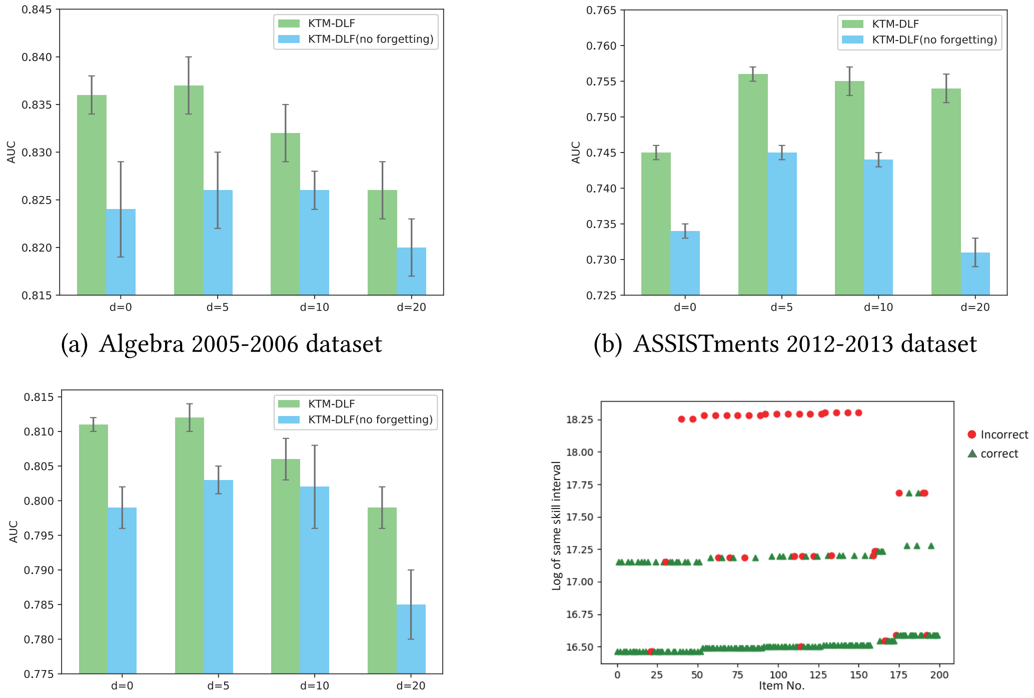
4.5.2 Effectiveness of the Dimension of Features

As shown in Tables 4.2, 4.3 and 4.4, the KTM-DLF model with d=5 performs best on all of the datasets, while the same model with d=20 performs worst except on ASSISTments 2012-2013. This indicates that a smaller multidimensional embedding and the pairwise interactions could improve the performance as compared to d=0 (logistic regression), but the impact of the dimensions of features appears to be very limited, a result that is consistent with existing work [29, 30].

4.5.3 Effectiveness of Cognitive Item Difficulty

To test the effectiveness of the proposed cognitive item difficulty, we conducted some ablation tests. We changed the item difficulty in Eq. 4.1 as $d(i, j, t) = \delta_j + \Sigma_{k \in KC(j))} \beta_k$, which does not consider the difficulty in terms of the cognitive challenge it presents to individual learner and assumes the difficulty level of an item is fixed, i.e. only consider the item-oriented part and do not consider the user-oriented part.

The results on three datasets are plotted in Figure 4.4(a), 4.4(b) and 4.4(c). We observe that considering user-oriented cognitive item difficulty rather than the item-oriented inherent difficulty of items and KCs generally performs better on all three datasets. Notably, we can see that using cognitive item difficulty is more effective for the ASSISTments 2012-2013 dataset (the average AUC gain is +0.009 for various values of d) than for the other two (the average AUC gains are +0.001 and +0.002, respectively). This is consistent with the characteristics of the three datasets. As the ASSISTments 2012-2013 dataset includes more users and fewer items than the other two, students are more likely to attempt the same items or skills. Hence the change on the cognitive item difficulty might be much bigger. We visualize the difficulty level of the first 50 problems attempted by the first 40 students in skill level in ASSISTments 2012-2013 dataset using Eq. 4.2, and we also plot the corresponding response outcome, as shown in Figure 4.4(d). We can see that students answering difficult problems are more likely to obtain the wrong answers for their attempts than when they attempt the easy ones, showing the effectiveness of the proposed adaptive item difficulty.



(a) Algebra 2005-2006 dataset

(b) ASSISTments 2012-2013 dataset

(c) Bridge to Algebra 2006-2007 dataset

(d)

Figure 4.5: AUC comparisons of the effectiveness of forgetting on three datasets. (a), (b) and (c) show the performance comparisons between KTM-DLF model and its ablated model without considering the forgetting procedure on Algebra 2005-2006, ASSISTments 2012-2013, and Bridge to Algebra 2006-2007 dataset, respectively. (d) shows the effect of lag time interval on the responses of a student on 200 problems in the Algebra 2005-2006 dataset.

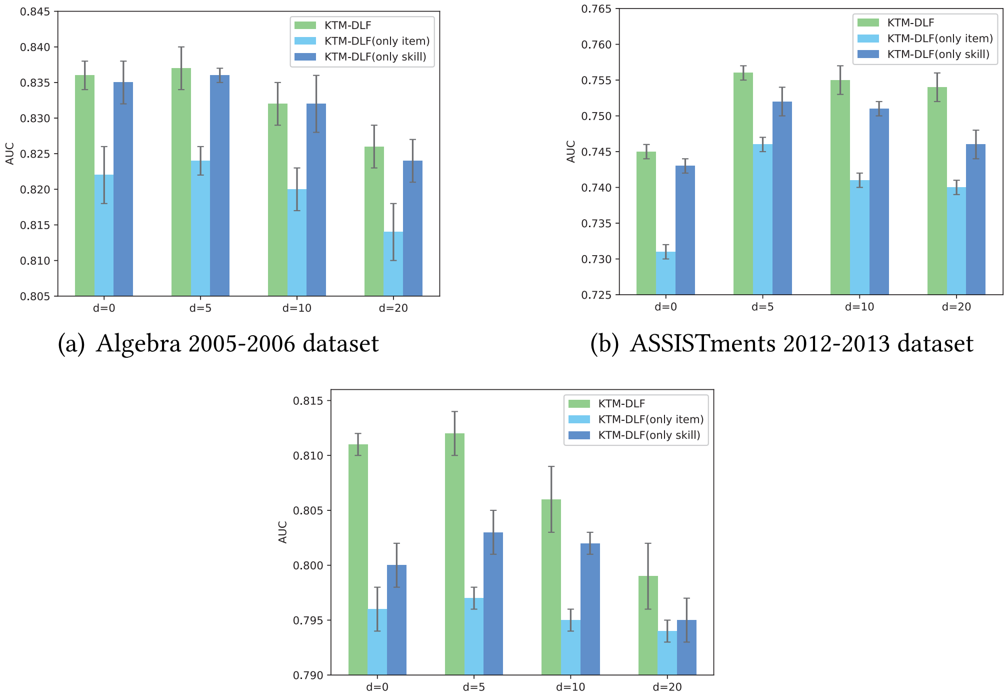
4.5.4 Effectiveness of Forgetting

To test the effectiveness of the forgetting procedure, we removed the term $f(i, j, t)$ in Eq. 4.6 and compared it with the full KTM-DLF models. The results on three datasets are shown in Figure 4.5(a), 4.5(b) and 4.5(c). We can see that the forgetting procedure shows +0.009 average AUC improvement on Algebra 2005-2006, +0.014 on ASSISTments 2012-2013 and +0.010 on Bridge to Algebra 2006-2007. Further, we plot the lag time of 200 problems attempted by a student in the Algebra 2005-2006 dataset in Figure 4.5(d). We can see that a long lag time interval from the same skills can lead to failure for the same practices in the later attempts, thereby verifying the effectiveness of the proposed forgetting procedure for predicting students' learning performance.

4.5.5 Effectiveness of Learning Measured by Item-only, Skill-only, and Item-Skill

To test the impact of the learning procedure measured by various factors, we conducted ablation tests by considering the learning procedure measured by item-only, skill-only, and item-skill. For item-only tests, Eq. 4.3 is changed to $l(i, j, t) = \Phi_{i,j,t}$. For skill-only tests, Eq. 4.3 is changed to $l(i, j, t) = \Sigma_{k \in KC(j))} \Phi_{i,k,t}$. The item-skill tests are based on the full KTM-DLF model.

The results on three datasets are shown in Figure 4.6. We observe that measuring students' learning by considering both items and skills associated with items in the history of their attempts obtains best performance, with the average AUC gains of +0.013 and +0.001 compared with the item-only and skill-only tests on Algebra 2005-2006, +0.013 and +0.005 on ASSISTments 2012-2013, and +0.012 and +0.008 on Bridge to Algebra 2006-2007 dataset. However, compared with the item-only tests, the skill-only tests perform better in predicting students' performance (this can be seen also from the above values of average AUC gains). This makes sense since the numbers of skills are quite limited compared with the numbers of items in all three datasets (112/173113, 265/52976 and 493/129263, respectively), hence introducing the item-level consideration might introducing more noise. Moreover, measuring students' learning procedure by considering skills compares more favorably to the proposed KTM-DLF model. In a cognitive model like KTM-DLF from systems like



(a) Algebra 2005-2006 dataset

(b) ASSISTments 2012-2013 dataset

(c) Bridge to Algebra 2006-2007 dataset

Figure 4.6: AUC comparisons of the effectiveness of the learning procedure measured by item-only, skill-only, and item-skill on three datasets.

Cognitive Tutor and ASSISTments, relying on skill makes sense. As shown in the previous discussion, solving an item need not only all the KCs involved in the item, but also some other factors, like the memory load imposed by the item representation and the search space to find the correct combination of KCs, hence taking both skill and item into consideration can model this effect and gain more information in the learning procedure, a reason why the KTM-DLF with learning measured by item-skill gets the best performance.

In general, from the experimental evaluation, we can see that the KTM-DLF model outperforms all of the other comparison baseline models and its ablated counterparts, verifying that modeling students' dynamic knowledge construction procedure and cognitive item difficulty can boost the performance of knowledge tracing over models that do not consider them or consider only one or the other.

4.6 Summary

This chapter focused on **Issue 1:** *what factors influence the learning performance and how to quantify these factors and utilize them to model the dynamic evolution of learner knowledge ?* We empirically examined both learner factors (learning and forgetting) and item factor (cognitive item difficulty) that influencing the learning performance, proposed methods to quantify these factors and utilized them into our proposed KTM-DLF model to assess the learner knowledge.

The KTM-DLF model traces the evolution of each student's knowledge acquisition to further predict his or her future performance by modeling the student's dynamic knowledge construction procedure and cognitive item difficulty. Specifically, this chapter first proposed the concept of cognitive item difficulty, which not only considers the difficulty from the items and KCs themselves, but also from the cognitive aspects of each student to make it user-oriented. Further it proposed a method to model cognitive item difficulty by making use of students' learning histories. The difficulty level of each item is calculated at both the item and KC levels, and it can also be adaptive to the individual student's cognition. Then, based on the learning and forgetting curve theories, it proposed methods to model these two dynamic procedures over time. Learning is modeled by leveraging the correct and incorrect attempts in different time windows at both the item and KC levels, and forgetting is modeled by considering the lag time from the previous interactions. Finally, it combined the above factors and used the factorization machine framework to enrich the proposed model. This framework can not only consider the contributions of different parts to the probability of observing binary outcomes of attempts (correct or incorrect), but also embed features in high dimensions and use the pairwise interactions to make the model more accurate. Extensive experiments were conducted on three public real-world datasets. The experimental results showed that the proposed model outperformed all of the other comparison baseline models and its ablated counterparts, verifying that modeling students' dynamic knowledge construction procedure and cognitive item difficulty all boosted the performance of knowledge tracing.

The proposed model has achieved good performance by taking consideration of various factors influencing the learning performance, however, it is based on a potential assumption that the items attempted by learners are independent to each other (i.e., the relationships among all the questions and skills are not considered in the model), which may lead to performance degradation on learner knowledge assessment because of the sparseness of response data and the potential information loss. We will further explore methods to alleviate this issue in the next chapter.

5 Knowledge Structure Enhanced Graph Representation Learning Model for Attentive Learner Knowledge Assessment

Recent learner knowledge assessment methods have achieved good performance at this task. However, the adequacy of KT is still challenged by the sparseness of the learners' exercise data. To alleviate the sparseness problem, most of the exiting studies implement their models based on the skills rather than the questions themselves, as questions are often numerous and associated with much fewer skills. However, at the skill level, KT neglects the distinctive information related to the questions

The material in this chapter is based on [152] themselves and their relations. In this case, the models can imprecisely infer the learners' knowledge states and might fail to capture the long-term dependencies in the exercising sequences. This is the **Issue 2** "*sparseness and information loss*" addressed in this thesis.

To alleviate this issue, this chapter explores to incorporate the knowledge structure (KS) into the learner assessment procedure to potentially resolve both the sparseness and information loss, an avenue not yet been fully explored because obtaining the complete KS of a domain is challenging and labor–intensive. Specifically, we propose a novel KS–enhanced graph representation learning model for KT with an attention mechanism (KSGKT). Section 5.1 introduces the motivation of our proposed solution in this chapter. Then the proposed KSGKT model is detailed in Section 5.2. Section 5.3 explains the experimental settings, and Section 5.4 presents the experimental results and analysis. Based on the KSGKT model, Section 5.5 shows a case study to provide the fine-grained diagnostic report to learners, and a summary of this chapter is given in Section 5.6.

5.1 Motivation

As we have discussed in the previous chapters, learner assessment task in an online learning system dynamically assesses the learner knowledge in a longitudinal manner. Based on the inferred knowledge states, learners are provided with various adaptive services that suit their individual needs, thereby improving their learning efficiency [23].

Sparseness and Information Loss Massive efforts have been devoted to track learner knowledge (also named Knowledge Tracing, KT) in the skill-level [41, 42], which build KT models based on the skills (or "knowledge concepts") required in a specific domain. Each question in a KT task is correlated with one or more skills needed to solve the question (e.g., "3+5" corresponds to the skill "addition of integers"), and each skill is related to many questions. The question–skill mapping information is typically encoded as a Q-matrix of prior knowledge provided by education experts [12, 33], and can be naturally represented as a question–skill relation graph (for an example, see Figure 5.1(left)). Most of the existing KT methods train the KT models on

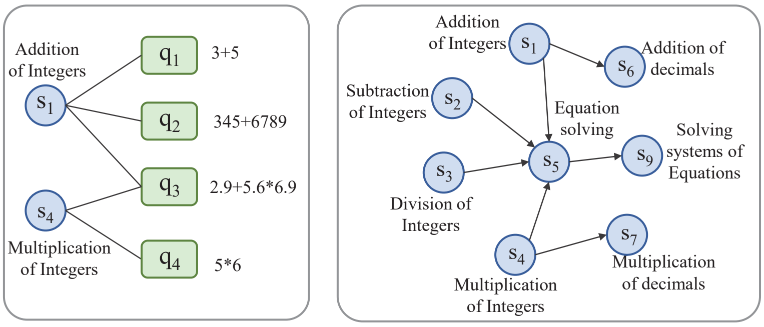


Figure 5.1: (left) Illustration of a question–skill relation graph, and (right) part of the knowledge structure in the algebra domain

skills rather than questions, as the number of questions is far greater than the number of skills and most students only attempt a small part of the questions. In general, students are not required to answer all the questions in an ITS, meaning that some students may not answer some questions. Accordingly, the response data are quite sparse [41, 42, 107]. This is also shown in Table 3.1, most of the datasets only have the sparseness value of less than 1%. Skill-level KT is feasible to some extent because skill mastery can largely affect the correctness of question answering. Therefore various KT methods proposed under this setting have achieved good performance.

In skill-level KT models, all questions pertaining to a specific skill are considered as the same inputs (and multiple skills corresponding to a question are merged into one new skill). This approach loses the distinctive information related to individual questions, leading to imprecise inferences of the learners' knowledge states [127, 126, 41, 42]. For example, in Figure 5.1(left), the questions "3+5" and "345+6789" both require the skill "addition of integers", and are considered as the same inputs when building the KT models, which ignores their different difficulty levels. Existing researches have proved that question difficulty undoubtedly influences the learner performance [12, 144, 93]. In this chapter, we explore to associate the cognitive question difficulty with the question representation, and introduce extra distinctive information for each question to potentially improve the reliability of tracing learner proficiency over the long-term.

The interdependencies between the skills in the knowledge structure (KS) have long been acknowledged in both cognitive science and artificial intelligence [107, 108]. The prerequisites between pedagogical concepts can be represented as a knowledge graph [53] (see Figure 5.1(right) for an example). Nevertheless, the KS (which specifies the relations among skills) has rarely been integrated in KT models because obtaining the complete KS of a domain is labor-intensive and the KS is not easily inferred from the data [109, 77, 128]. To avoid these difficulties, most of the KT models simply assume that all questions and skills are independent. In the real world, questions are interrelated to each other and are also closely related to the underlying skills required for their solution. When learners grow their knowledge from a certain question incorporating $skill_1$, they also improve their attainment of $skill_2$ to some extent (assuming that $skill_1$ is related to $skill_2$ in the KS). For example, a learner who attempts an exercise requiring the "equation solving" skill will also deepen his or her understanding of the skill "solving systems of equations". In this work, we infer the KS from learners' response data and integrate it with the KT model. Our method offers two advantages over the existing models: first, it inputs extra information into the question representation by referencing the KS (thus alleviating the data sparsity problem in the question representation), and second, it models the impact of previous experiences on future exercise during the knowledge evolution. In addition, incorporating the KS into the KT procedure can capture the long-term dependencies in the exercising sequences [31], further improving the precision of inferring the dynamic knowledge proficiencies of learners.

Moreover, most of the previous work on KT represents the questions for model building using one-hot encoding [10]. The resulting data are often too sparse to represent sufficient information for the KT task, thus leading to performance degradation [41, 51, 42, 128]. In recent work on graph representation learning, the model is trained on a dense embedding of the graph, which improves the performance on various tasks [153, 154]. Motivated by the high ability of graph neural networks to extract graph representation by aggregating the information from neighbors, we apply the graph representation learning method to obtain question- and skill-embedding from the KS enhanced question–skill relation graph. The learned embeddings from the graph incorporate not only the explicit multi-hop question–skill relations but also the implicit multi-hop question–question and skill–skill relations in the graph. We also propose a convolutional representation method that incorporates additional information and considers their interactions, thus generating dense and meaningful representations of the input questions and potentially further improving the model performance.

Overview of our Solution We propose a new KT model named Knowledge Structure enhanced Graph Knowledge Tracing (KSGKT) that traces the evolution of learners' knowledge proficiencies and solves the issue of data sparseness and information loss. We first explore eight methods to infer the domain KS from the learner response data and integrate it with the original question–skill relation graph to obtain the KS enhanced question–skill relation graph. Leveraging a graph representation learning model (Metapath2Vec [153]), we then obtain the dense question and skill embeddings from the enhanced graph. To overcome the limitations of skill–level KT models, which neglect the distinctive information related to the questions, we propose a convolutional representation method that integrates the question and its associated skill embedding information into the multi-level cognitive question–difficulty information, thus obtaining a comprehensive representation of each attempted question. These representations for the learners' exercising sequences are fed into the proposed KT model, which considers the long-term dependencies using an attention mechanism, and finally predicts the learners' performance on new problems. The main contributions of this work are listed below.

- We propose the KSGKT model enhanced by the inferred KS in the domain that traces the evolution of learners' knowledge proficiencies with three attention methods.

- We explore eight methods that automatically discover the domain KS from learner response data, and test them in the KT procedure.

- We propose a KS–enhanced graph representation learning model that learns the dense question and skill embeddings in the KS enhanced graph, and a convolutional representation method that fuses these distinctive heterogeneous features into a comprehensive question representation.

- We conduct comprehensive experimental evaluations from six perspectives on three real-world datasets. The results demonstrate the superiority of our method in dynamically modeling the learning performance and discovering the KS from data.

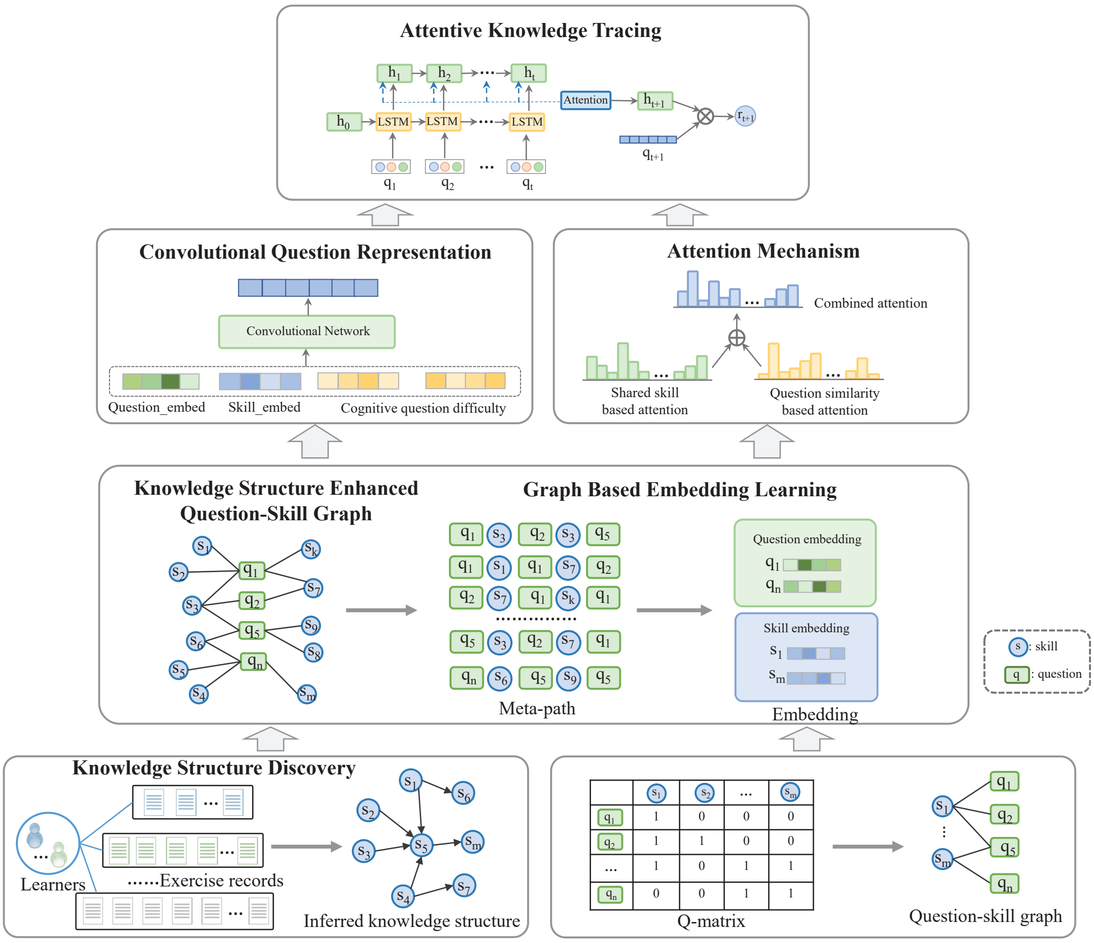

Attentive Knowledge Tracing
Attention
LSTM
Convolutional Question Representation
Convolutional Network
Question_embed
Skill_embed
Cognitive question difficulty
Attention Mechanism
Combined attention
Shared skill based attention
Question similarity based attention
Knowledge Structure Enhanced Question-Skill Graph
Graph Based Embedding Learning
Meta-path
Question embedding
Skill embedding
Embedding
: skill
: question
Knowledge Structure Discovery
Learners
Exercise records
Inferred knowledge structure
Q-matrix
Question-skill graph

Figure 5.2: The proposed KSGKT framework for attentive knowledge tracing

5.2 KSGKT Model

This section introduces the proposed KSGKT framework for knowledge tracing, as shown in Figure 5.2. The framework proceeds the KT task using five modules: knowledge structure discovery, graph-based embedding learning, convolutional question representation, attention mechanism and learner knowledge state evolution. Before conducting the embedding learning, we must build the KS–enhanced question–skill graph by integrating the KS inferred from the learner response data into the question–skill graph. Based on the built KS–enhanced question–skill graph, the dense embeddings of all skill and question nodes are obtained through the graph embedding learning method Methpath2Vec. To incorporate more distinctive information of

Table 5.1: A list of symbol notations used in this study.

Var.	Description
u_i	learner u_i
q_t	question q_t
s_k	skills s_k
r_t	the correctness of the learner's answer at timestamp t
et_t	the elapsed time spent on solving the given questionat timestamp t
O	the binary q-matrix
R^w	the representation of the KS as a skill relation matrix
$\hat{R}$	matrix representation of the KS–enhanced question–skill graph
q	question embedding vector
$\bar{s}$	average skill embedding vector in a question
d_q	question difficulty embedding vector at the question level
d_s	question difficulty embedding vector at the skill level
$\widetilde{q}$	the final comprehensive question embedding
x_t	the interaction embedding at timestamp t
h_t	the hidden knowledge state of a learner at step t
$\alpha_{i,t+1}$	the attention between current questions and previous question
$g(\widetilde{q}_i, \widetilde{q}_{t+1})$	the correlation between current questions and previous question
p_{t+1}	the predicted learner performance at step $t+1$

the questions, the cognitive question difficulty at both the question and skill levels are also inferred from the learning histories of individual learners. A convolutional representation method is then proposed to fuse the question and skill embeddings with the cognitive question difficulty. It considers each separate factor and the interactions between each pair of factors, thus obtaining the comprehensive representations of questions (see Figure 5.3). These representations are fed to the attentive KT network for predicting learner performance (see Figure 5.4 for the KT procedure). To capture the long-term dependencies in the exercising sequence, different attention mechanisms are applied. For reference, the algorithm for our proposed method is shown in Alg. 5.1. A list of symbol notations used in this study is presented in Table 5.1 to facilitate reading.

Algorithm 5.1 The proposed KSGKT method for attentive knowledge tracing

Input: the learners' exercise logs E in a system, the Q-matrix O

Output: the predicted probability p_{t+1} that the learner answers exercise q_{t+1} correctly, the matrix representation R^w of the KS, and the final embedding matrix Q, S of the question and skill nodes.

```
$R^w, \hat{R} \leftarrow \mathbf{0}$, $\alpha_{i,t+1}, p_{t+1} \leftarrow 0$;
initialize $Q, S, D$ from a Gaussian distribution;
infer $R^w$ from the exercise logs $E$;    ▷ Eight methods in Section 5.2.1
calculate $\hat{R}$ using Eq. 5.3 and 5.4;
calculate $\Psi_{q,t}$ and $\Psi_{s,t}$ using Eq. 4.2;    ▷ Cognitive item difficulty
repeat
    $Q, S \leftarrow Metapath2Vec(\hat{R}$,"QSQ", $Q, S)$;    ▷ Embedding learning in Section 5.2.2
    $\widetilde{q} \leftarrow Convolution(q = Q_q, s = avg(S_q), d_q = \Psi_{q,t}D, d_s = \Psi_{s,t}D)$;    ▷ Convolutional question representation in Section 5.2.3
    $h_t \leftarrow LSTM(Concat(\widetilde{q}, t_t, r_t), h_{t-1}; \theta)$;    ▷ update learner knowledge state using LSTM in Section 5.2.4
    calculate attention $\alpha_{i,t+1}$ using Eq. 5.14;
    $p_{t+1} = MLP(\widetilde{q_{t+1}}, h_i, \alpha_{i,t+1})$;    ▷ Predict learner performance with three attention methods in Section 5.2.4
until $\mathcal{L} = -\sum_t (r_{t+1} log\, p_{t+1} + (1 - r_{t+1}) log(1 - p_{t+1}))$ is minimum    ▷ minimize cross-entropy loss
return $p_{t+1}, R^w, Q, S$.
```

5.2.1 Inferring Knowledge Structure from Data

In practical educational scenarios, there always exists a topological order (KS) among the skills in a domain, because skills are taught and learned in sequence. In many learning systems the KS information is never provided and must be obtained by time-consuming labor. We observed that if learners have not mastered a specific skill, their probability of incorrectly answering questions requiring the post-requisite skills will increase. Moreover, learners' mastery of a skill can be indicated by their performances on the attempted questions requiring that skill. Based on these observations, this chapter aims to discover the KS from the learner response data.

Previous studies [110, 77] intuited that the KS is difficult to directly extract from response data, but this difficulty can be bridged by ordering the learners' mastery of skills. Hence, in this work, we infer the KS from the order of the learners' mastery of

Table 5.2: Contingency table for a pair of skills s_i and s_j

	s_j master	s_j not master	total
s_i master	a	b	a+b
s_i not master	c	d	c+d
total	a+c	b+d	a+b+c+d

skills, which is explicitly represented by the exercising performance data. Here we first give a formal definition of the KS.

Definition 1 (Knowledge structure): The KS is a directed graph with all skills as nodes. It can be represented as $KS = \left\{S, \overrightarrow{L}\right\}$, where S is the set of all skill nodes and $\overrightarrow{L}$ denotes the prerequisite relations between two skill nodes in the graph. KS can also be represented as a skill relation matrix $R \in \mathbb{R}^{|S| \times |S|}$, in which entry $R_{i,j}$ represents the prerequisite relation $s_i \rightarrowtail s_j$ between skill s_i and s_j.

Inspired by the definition of question similarity in previous methods [155, 51, 102], we explore the underlying KS using eight methods: Skill Transition, Cohen's Kappa, Adjusted Kappa, Phi coefficient, Yule coefficient, Ochiai coefficient, Sokal coefficient, Jaccard coefficient, as described below.

- **Skill Transition**: The skill-transition matrix R contains the transitions of different skills. Its entries are $R^{SK}_{i,j} = \frac{n_{i,j}}{\sum_{k=1}^{|\kappa|} n_{i,k}}$, where $n_{i,j}$ denotes the number of times in which skill s_j is trained immediately after training skill s_i.

To further leverage the impact of the learners' performance of one skill on the performance of another, we summarized the learners' performance on skill pair s_i and s_j in a contingency table (see Table 5.2). As mentioned above, we interpreted the learners' correct or incorrect responses as mastery indicators of the underlying skills of the given questions. It is worth noting that in Table 5.2, the question requiring skill s_i occurs before the question requiring s_j in the learning sequence. When there are multiple occurrences of question pairs in the learning sequence, we consider only the latest occurrence. Based on the contingency table, we discovered the KS using the evaluation indices [102] in Table 5.3, which measure the agreement of the prerequisite relation between a pair of skills. These indices are widely used for measuring associations between two variables.

Table 5.3: Evaluation indices for obtaining KS using the contingency table

Cohen's Kappa	$R_{i,j}^{Kappa} = 2(ad - bc)/\{(a+b)(b+d) + (a+c)(c+d)\}$
Adjusted Kappa	$R_{i,j}^{Kappa'} = 2(ad - bc)/\{(a+c)(c+d)\}$
Phi coefficient	$R_{i,j}^{Phi} = (ad - bc)/\sqrt{(a+b)(b+d)(a+c)(c+d)}$
Yule coefficient	$R_{i,j}^{Yule} = (ad - bc)/(ad + bc)$
Ochiai coefficient	$R_{i,j}^{Ochiai} = a/\sqrt{(a+b)(a+c)}$
Sokal coefficient	$R_{i,j}^{Sokal} = (a+d)/(a+b+c+d)$
Jaccard coefficient	$R_{i,j}^{Jaccard} = a/(a+b+c)$

As the KS is always a unidirectional graph, we simplified it by a suitable strategy. We also imposed a threshold that controlled the sparsity of the relations in KS. The final skill relation matrix was denoted as R^w, $w \in \{SK, Kappa, Kappa', Phi, Yule, Ochiai, Sokal, Jaccard\}$. The elements along the diagonal of R^w were set to one.

$$\begin{cases} R_{i,j}^w = max(R_{i,j}^w, R_{j,i}^w), R_{j,i}^w = 0, & if R_{i,j}^w \geq R_{j,i}^w \\ R_{j,i}^w = max(R_{i,j}^w, R_{j,i}^w), R_{i,j}^w = 0, & otherwise \end{cases} \tag{5.1}$$

$$R_{i,j}^w = \begin{cases} 1, & if R_{i,j}^w \geq threshold \\ 0, & otherwise \end{cases} \tag{5.2}$$

5.2.2 Embedding Learning on the KS–enhanced Question–Skill Graph

Graph representation learning models such as graph neural networks (GNNs) have solved various tasks through their excellent ability to process graph-structure data. Traditional machine learning methods have limited ability to extract and encode the high-dimensional, non-Euclidean information on graph structure from graphs [156]. Following the edges in the graph, GNNs obtain the node representation from a whole graph by propagating and aggregating information from the neighbor nodes. In this way, node embeddings can summarize their graph positions and the structures of their local-graph neighborhoods [156]. The obtained graph-based embeddings can be directly used for various downstream tasks. In the deep-learning models of

KT tasks, they alleviate the sparsity problem of the one-hot representation. One widely used GNN model is the graph convolutional network (GCN), which was recently used as the input-question embedding learning in KT task [41, 51]. GCNs are especially suitable for isomorphic graphs, in which the nodes are of the same type. Recent graph-to-vector models such as Metapath2Vec [153] have proven successful in heterogeneous graph-representation learning. Such models are eminently suitable for KT tasks because question–skill graphs are typical heterogeneous graphs. In this work we leverage the Metapath2Vec model to learn the dense embedding from the relation graph. Before presenting the embedding learning process, we now introduce some definitions that will used in this sections.

Definition 2 (Question-skill graph): The q-matrix O containing the question–skill relations can be naturally represented as a question–skill graph $G = \{Q, S, L\}$, where Q and S are sets of question and skill nodes, respectively, and $L = [O_{jk}] \in \{0, 1\}$ indicates whether question node q_j and skill node s_k are connected by an edge.

Definition 3 (KS Enhanced Question-skill graph): KS–enhanced question–skill graph is a dense graph, with all questions and skills as nodes. It can be represented as $KSG = \left\{Q, S, L_{q,s}, L_{s,s}\right\}$, where $L_{q,s} = [O_{q,s}] \in \{0, 1\}$ indicates the original question–skill relation in the Q-matrix and $L_{s,s} = [R^w_{s,s}] \in \{0, 1\}$ denotes the prerequisite relations between two skill nodes in the KS. It can be also represented as a matrix $\hat{R} \in \mathbb{R}^{|Q|\times|S|}$.

As we have already obtained the inferred skill relation matrix $R^w \in \mathbb{R}^{|S|\times|S|}$ of the KS and the q-matrix $O \in \mathbb{R}^{|Q|\times|S|}$ for the question–skill graph, we now built a KS–enhanced question-skill graph and conducted embedding learning on it.

The matrix representation $\hat{R} \in \mathbb{R}^{|Q|\times|S|}$ of the KS–enhanced question–skill graph is obtained as follows:

$$\hat{R} = O(R^w)^T \tag{5.3}$$

$$\hat{R}_{i,j} = \begin{cases} 1, & if \hat{R}_{i,j} \geq 1 \\ 0, & otherwise \end{cases} \tag{5.4}$$

In Eq. (4), the transpose of the skill relation matrix R^w accounts for the skills that are prerequisite to the skills of the current question. In other words, a question requiring a specific skill is also related to the prerequisite skills.

The matrix $\hat{R}$ can be naturally represented as a graph, hereafter denoted as the

KS-enhanced question–skill graph. This graph includes not only the multi-hop relations between questions and skills, but also the prerequisite relations among the skills. The embeddings in this graph were obtained using the Metapath2Vec method [153].

The Metapath2Vec method proceeds in two main steps: meta-path generation and skip-gram-based embedding learning. A meta-path is a sequence of nodes following the edges in the graph. To assure that all questions and skills appear in the final embeddings, we generate the meta-paths from the KS-enhanced question–skill graph using a question–skill–question (QSQ) pattern, in which every meta-path begins with a question node followed by a skill node and then by a question node; for example, ρ: $q_1 \overset{\hat{R}_{1,1}}{\longmapsto} s_1 \overset{\hat{R}_{2,1}}{\longmapsto} q_2 \overset{\hat{R}_{2,2}}{\longmapsto} s_2 \overset{\hat{R}_{3,2}}{\longmapsto} q_3$. Setting two hyper-parameters—the path length $\wp$ and number of paths $\aleph$—for each question node, we generated all meta-paths on the graph. The probability of moving one step between two nodes v_t and v_{t+1} along path ρ is given by

$$P(v_{t+1}|v_t, \rho) = \begin{cases} 1/|N_t(v_t)|, & if v_{t+1} \in N_t(v_t), type(v_t, v_{t+1}) \in \text{“}QSQ'' \\ 0, & otherwise \end{cases} \tag{5.5}$$

where $N_t(v_t)$ is the set of one-hop neighbors of nodes v_t following the QSQ pattern.

Following [153], we apply the heterogeneous skip-gram and learn the node embeddings by maximizing the probability of having context $N_t(v_t)$ given a node v_t:

$$argmax_\theta \sum_{v \in V} \sum_{type \in \text{“}QSQ''} \sum_{c_t \in N_t(v_t)} logp(c_t|v_t, \theta) \tag{5.6}$$

where $p(c_t|v_t, \theta)$ usually defined as a softmax function based on the learned embedding of nodes. Finally, we obtain all embeddings with the same dimension d of question and skill nodes in the graph, which also enclose the relation information.

5.2.3 Convolutional Question Representation

In this subsection, we fuse the various feature representations through convolutional operations to obtain the comprehensive question embedding with distinctive information.

We have now obtained the distinctive embedding and the contained skill embedding

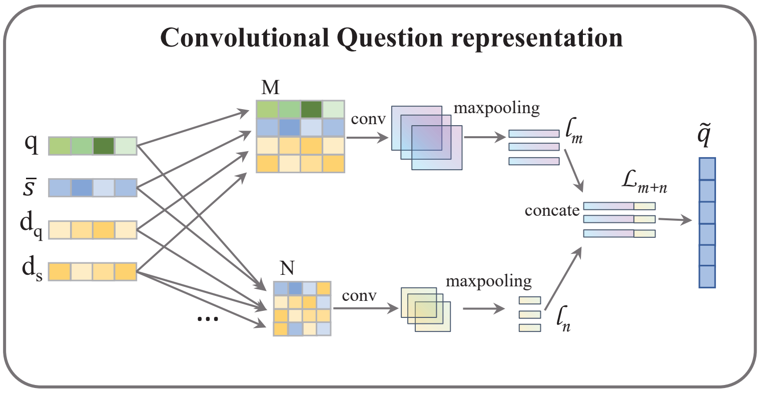


Figure 5.3: Use of convolution to fuse various distinctive features and their interactions into the comprehensive question representation

in the previous subsection. To include more distinctive information into the question representation, we also calculate the cognitive difficulty information of each question (see Section 4.3.1). To simultaneously preserve all of these parts, we fuse them into comprehensive question embeddings. Following [42] and [157] that learn the high-order latent patterns through feature interactions and convolution operations (rather than directly concatenating the features), we map and fuse the separate features and their interactions using convolution operations, as shown in Figure 5.3.

The question-difficulty information is represented as vectors using an embedding matrix D (of size $(c + 1) \times d$). The continuous embedding vectors at the question and skill levels of each question are defined as $d_q = \Psi_{q,t} D$ and $d_s = \Psi_{s,t} D$, respectively. Note that $\Psi_{q,t}$ and $\Psi_{s,t}$ are calculated using Eq. 4.2. For questions containing multiple skills, we represent the skill as the average skill embedding as

$$\bar{s} = \frac{1}{|s_q|} \sum_{s_i \in s_q} s_i. \tag{5.7}$$

Fusing the above-obtained features, we generate the linear information M and quadratic information N for question q.

$$\begin{aligned} M &= [q, \bar{s}, d_q, d_s] \in \mathbb{R}^{4 \times d}, \\ N &= [\langle M_i, M_j \rangle] \in \mathbb{R}^{4 \times 4}, \end{aligned} \tag{5.8}$$

where $\langle . \rangle$ represents the interactions of two vectors obtained by the inner product. We then apply the two-dimensional convolution operation with eight kernels of size 2×2 on both M and N, and maxpooling on each feature map to obtain $l_m \in \mathbb{R}^{1\times(d-1)\times 8}$ and $l_n \in \mathbb{R}^{1\times 3\times 8}$. These two parts are then concatenated into eight longer vectors $l_{m+n} \in \mathbb{R}^{1\times(d+2)\times 8}$ including the convolutions from the separate features and their interactions. Inspired by the multi-head mechanism in the transformer model [125], we concatenate and linearly transform the eight vectors and hence obtain the final question representation $\widetilde{q} \in \mathbb{R}^{1\times d'}$.

$$\begin{aligned} l_m &= MaxPooling(Conv(M)), \\ l_n &= MaxPooling(Conv(N)), \\ l_{m+n} &= Concat(l_m, l_n), \end{aligned} \tag{5.9}$$

$$\widetilde{q} = Concat(l^1_{m+n}, ..., l^8_{m+n})W^O, \tag{5.10}$$

where $W_O \in \mathbb{R}^{((d+2)\times 8)\times d'}$ is the parameter that transforms the convolution results into a vector.

5.2.4 Learner Knowledge State Evolution

The learner exercising sequences are fed into an attentive KT framework that predicts the learner performance, as shown in Figure 5.4.

The log data of each interaction in the exercising sequences consists of a tuple representing the question, the correctness, and the elapsed time. Look-up operations are performed on an embedding matrix $E_r \in \mathbb{R}^{2\times d'}$, in which row vector r_t contains the incorrectness or correctness of the responses. The elapsed time et strongly evidences a student's proficiency in knowledge and skills [158]. This time is converted to seconds and capped at 500 seconds. A d'-dimensional latent embedding vector for et_k is computed as $t_k = et_k W_{et} + b_{et}$, where W_{et} and b_{et} are learnable vectors. The interaction embedding is obtained as

$$x_t = Concat(\widetilde{q}, t_t, r_t). \tag{5.11}$$

The sequence data of the learners' exercising process are modeled using LSTM [10]

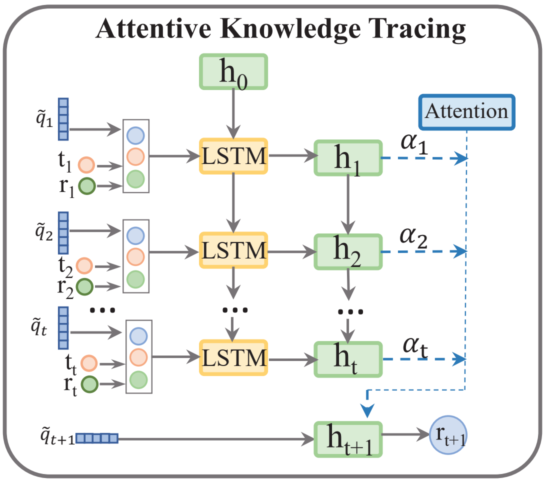


Figure 5.4: Tracking the evolution of a learner's knowledge state by attentive knowledge tracing on the exercising sequences

within the KT framework, and the learners' knowledge states are traced at each time. The hidden knowledge state h_t of a learner at step t is updated based on the current input and the previous state $h_t = LSTM(x_t, h_{t-1}; \theta)$ as:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i),\\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f),\\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o),\\
c_t &= f_t c_{t-1} + i_t tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c),\\
h_t &= o_t tanh(c_t),
\end{aligned} \tag{5.12}$$

where the i_t, f_t, o_t are the input, forget, and output gates, respectively, c_t is the cell memory vector, and W_* and b_* are network parameters.

We then employ an attention mechanism that accounts for the impact of previous attempts on the current attempt. A new question will likely be strongly affected by similar questions or questions requiring the same skillset as the new question. Moreover, according to forgetting-curve theory, the impact exponentially decays as time passes [55]. To describe these effects, we assume that the learner-knowledge state in the current step is the weighted sum of the aggregated states in the previous steps.

The weights are based on the correlations:

$$h_{t+1} = \sum_{i=1}^{t} \alpha_{i,t+1} h_i. \tag{5.13}$$

The attention was calculated by three methods: shared skill-based attention, question similarity-based attention, and a combination of both former attention methods.

$$\begin{aligned} \alpha_{i,t+1} &= \frac{exp(corre_{i,t+1})}{\sum_j exp(corre_{j,t+1})}, \\ corre_{i,t+1} &= exp(-\theta|t_{t+1} - t_i|)g(\widetilde{q}_i, \widetilde{q}_{t+1}), \end{aligned} \tag{5.14}$$

where $|t_{t+1} - t_i|$ is the time interval between step i and $t + 1$, $\theta > 0$ is the learnable decay rate over time, the exponential term down-weights the importance of questions in the distant past, and $g(.)$ denotes the correlation between two questions. The $g(.)s$ computed by the three methods provide three kinds of attentions.

- **shared skill based-attention**: This method calculates the number of skills shared by two questions in the KS-enhanced q-matrix $\hat{R}$. We define $g(\widetilde{q}_i, \widetilde{q}_{t+1}) = \frac{n}{|s_{q_{t+1}}|}$, where n is the number of shared skills and $|s_{q_{t+1}}|$ denotes the number of skills contained in question q_{t+1}.

- **question similarity-based attention**: This method calculates the correlations between two question representations by determining their similarities. We define $g(\widetilde{q}_i, \widetilde{q}_{t+1}) = cos(\widetilde{q}_i, \widetilde{q}_{t+1})$, that is, the cosine similarities between the two vectors of question representation.

- **combined attention**: This method combines the above correlations with coefficients. We define $g(\widetilde{q}_i, \widetilde{q}_{t+1}) = \lambda \frac{n}{|S_{q_{t+1}}|} + (1 - \lambda)cos(\widetilde{q}_i, \widetilde{q}_{t+1})$, where λ is a tunable parameter that balances the above two correlations.

The learner performance at step $t + 1$ can be predicted from the question representation $\widetilde{q}_{t+1}$ and the current knowledge state h_{t+1} as follows:

$$\begin{aligned} s_{t+1} &= tanh(W_s[\widetilde{q}_{t+1}, h_{t+1}]) + b_s, \\ p_{t+1} &= \sigma(W_p s_{t+1} + b_p), \end{aligned} \tag{5.15}$$

where W_* and b_* are parameters in the fully connected layer and the sigmoid activation layer, respectively.

Finally, we optimized our model by the cross-entropy loss; specifically, we minimized the following objective function between the true answer l_t and the predicted performance p_{t+1} at each interaction:

$$\mathcal{L} = -\sum_t (r_{t+1} log\, p_{t+1} + (1 - r_{t+1}) log(1 - p_{t+1})). \tag{5.16}$$

5.3 Experimental Settings

In a series of experimental tasks, we evaluated the proposed KSGKT model on three public real-world datasets. This section describes the experimental settings (aims, datasets, comparison baselines, setup and implementation, and evaluation metrics). The detailed experimental results and model analysis are presented in the next section.

5.3.1 Experimental Aims

Our experiments aimed to answer the following questions:

1. Based on the knowledge proficiency inferred from the learners' exercise histories, how well does KSGKT predict the learners' performance on new questions? (See subsection 5.4.1)

2. How does embedding learning on the KS-enhanced graph affect the performance of the proposed model? (See subsection 5.4.2)

3. Does the question embedding learned on the graph provides meaningful information? (See subsection 5.4.3)

4. How well do the eight methods infer the KS from data? Is the inferred KS explainable? (See subsection 5.4.4 and 5.4.5)

5. How effective is the convolutional question representation of the KT task? (See subsection 5.4.6)

6. How effective is the attention mechanism in the proposed method? What is the effect of changing the attention-calculating method? (See subsection 5.4.7)

5.3.2 Datesets and Compared Models

Three well-established datasets are used to perform the experiments: Assist0910, Assist1213, and EdNet. The detail of these datasets are shown in Section 3.2. As part of our model evaluation, we competed the model against several state-of-the-art skill– and question–based KT models. The performances of the various models in different settings were compared using AUC as the evaluation metric.

- **BKT** [9] conducts KT using a hidden Markov model, and represents the learner knowledge states as a set of binary variables.
- **DKT** [10] is the first KT model based on a deep neural network. This model treats the learner-knowledge prediction as a sequence learning task and captures the complex representations of student knowledge using the hidden vectors of RNNs. The input is a one-hot encoding of skills.
- **DKVMN** [32] establishes the learners' knowledge states using an auxiliary memory that augments the neural networks. This method embeds the skill information into a key matrix and accumulates the temporal information from the learners' exercising sequences. It then infers their knowledge states on these skills.
- **KTM** [30] is a factor analysis model based on the factorization machine. It is a generic framework that incorporates the side information into the student model. Learner performance is predicted based on a sparse set of weights applied to all features in the samples.
- **DKT-Q** is a variant of DKT that replaces the skills embedding with a one-hot encoding of questions as the input.
- **DKT-Q&S** is a variant of DKT that inputs both the questions and skill representations to the DKT.
- **DKT-CQE** is a variant of DKT that inputs our convolutional question embeddings to the DKT.
- **GIKT** [41] is a graph-based interaction model that learns the question representations from the question–skill relation graph using a GCN and conducts KT within the LSTM framework.

Among the baseline models, the former three are skill-based models, in which the KT

Table 5.4: Comparisons of different learning models

Category	Representative Work	Typical Technique	Data Source					Use of Skill	Consider Sequence Order
			Problem/Skill Representation	Difficulty	Q-S Relation	KS	Elapsed time		
Probabilistic Model	BKT	HMM, Bayesian Network	parameters	–	×	×	×	single	✓
Factor Analysis Model	KTM	Factorization Machine	parameters	skill-level, constant	×	×	×	multiple	×
Deep Learning Model	DKT, DKVMN	RNN, LSTM, MANN	one-hot encoding	–	×	×	×	single	✓
Graph-based Model	GIKT	GCN	dense embedding	constant	✓	×	×	multiple	✓
	Our model	Metapath2Vec	convolutional dense embedding	multi-level, adaptive	✓	✓	✓	multiple	✓

is based on the skills contained in the questions. The latter five and the proposed model are question-based models that account for the distinctive question information. The different types of models are compared in Table 5.4.

5.3.3 Setup and Implementation

Before conducting the experiments, we extracted 20% of the sequences in the dataset as the test set and retained the remaining 80% as the training set.

To embed the nodes in the graph using Metapath2Vec, we set the length of all meta-paths as $\wp = 7$ and the number of paths as $\aleph = 100$ for each question node in the graph. The embedding dimension d of the skill and question representations was set to 128. The final dimension of the convolutional question representation was $d^{'} = 256$. The size of the hidden layers of the LSTM was set to 256. The other hyperparameters were set through grid searching. The embedding matrix of the correctness and all parameter matrices in the networks were randomly initialized and updated through the training process.

The model was optimized using Adam optimization of the learning rate on a case-by-case basis in the three datasets. The norm clipping threshold and batch size were maintained at 10 and 64, respectively. Similarly to the existing models, the sequence length of the model input was fixed at 200. Accordingly, the long sequences were divided into several short sequences and the short sequences were padded with null symbols to extend their length to 200.

The proposed model was implemented using TensorFlow. Our model was tested over 50 epochs because it converged over that period. The other baselines were implemented with their best parameter settings, as specified in the original works.

5.4 Results and Analysis

This section presents the experimental results and model analysis. To answer our first research question in subsection 5.3.1, subsection 5.4.1 compares the learner-score prediction performances of our proposed KSGKT model and all baseline models. The other five questions are answered in relevant model analyses. In subsection 5.4.2, the contribution of the embedding learning on the KS-enhanced graph is evaluated in an ablation study and the second question is answered. Subsection 5.4.3 presents the visualization of the question embedding learned on the graph and answers the third question. Subsection 5.4.4 and 5.4.5 compare the eight methods for KS discovery and verify the interpretability of the inferred KS, and the fourth question is answered. To answer the fifth and sixth research questions, the contributions of two separate components (convolutional question representation and the attention mechanism) are evaluated through the ablation studies in subsection 5.4.6 and 5.4.7.

5.4.1 Performance Prediction

The different models were evaluated by their performances in predicting the future learner scores from the estimated knowledge state. Table 5.5 presents the AUC results of all models on the three datasets.

Our model outperformed the other models on all three datasets. Specifically, the AUC scores of the KSGKT model were 0.8242, 0.7851, and 0.7754 on the Assist0910, Assist1213 and EdNet datasets, respectively, 3.97%, 1.39%, and 2.25%, respectively, above those of the state-of-the-art GIKT model. Similarly to the original DKT model, our model processes the time-series data using a recurrent neural network framework, but achieved 8.3%, 5.95% ,and 8.65% higher AUCs than the DKT model on the Assist0910, Assist1213 and EdNet datasets, respectively.

The skill-based BKT model was the worst performer among the models because it tracks the mastery of each skill separately, without considering a contextual trial sequence of all skills. DKT and DKVMN achieved similar performances and

Table 5.5: Comparisons of the AUC results of different models on the three datasets

	Model	ASSIST0910	ASSIST1213	EdNet
Skill-based Model	BKT	0.6571	0.6204	0.6027
	DKT	0.7412	0.7256	0.6889
	DKVMN	0.7559	0.7247	0.6921
Question-based Model	KTM	0.7582	0.7212	0.6899
	DKT-Q	0.7306	0.7158	0.6812
	DKT-Q&S	0.7616	0.7389	0.7235
	DKT-CQE	0.7998	0.7686	0.7523
	GIKT	0.7845	0.7712	0.7529
	KSGKT	**0.8242****	**0.7851***	**0.7754***

considerably outperformed the BKT model, confirming the effectiveness of applying deep neural networks to this task; however, they were slightly outperformed by the other question-based models. KTM framework, which incorporates several traditional models, typically obtained similar AUC scores to those of DKT and DKVMN. DKT extended with various input-question embeddings (DKT-Q, DKT-Q&S, and DKT-CQE) demonstrated noticeable performance differences. DKT-Q using the one-hot encoding of question representations performed much worse than the original DKT model, owing to the sparsity of question interactions in these datasets. DKT-Q&S and DKT-CQE decidedly outperformed the original DKT and DKT-Q models, consistent with our intuition that each question contains distinctive information even when it requires the same skills as one or more other questions in the dataset. Therefore, incorporating the distinctive question and skill information into the question representations can improve the model performance. The comparison between DKT and DKT-CQE also shows the effectiveness of the proposed convolutional question representation. Moreover, the improvements in the AUC scores of these DKT-extended models were smaller on Assist1213 than on the other two datasets. As Assist1213 is a single-skill dataset, incorporating the skill information into the question representation provides less additional information on this dataset than on the other datasets. Our KSGKT model outperformed GIKT, validating the effectiveness of the proposed KS-enhanced

Table 5.6: Comparisons of different question-embedding methods in different models on the three datasets. The best results are marked with *.

Methods	ASSIST0910	ASSIST1213	EdNet
DKT	0.7412	0.7256	0.6889
DKT-Q	0.7306	0.7158	0.6812
DKT-Q&S	0.7616	0.7389	0.7235
DKT-CQE	0.7998	0.7686	0.7523
GIKT	0.7845	0.7712	0.7529
KSGKT-Q	0.7409	0.7277	0.6987
KSGKT-S	0.7523	0.7399	0.7043
KSGKT-Q&S	0.7682	0.7464	0.7293
KSGKT-CQE	0.8242*	0.7851*	0.7754*

question–skill graph learning method.

5.4.2 Effect of Embedding Learning on Graph

Besides the aforementioned three extensions of DKT models, we extended our KSGKT model with the following question embeddings:

- **KSGKT-Q**: In this embedding, we removed the module of embedding learning on the graph and inputted the one-hot question embedding to our KSGKT model.
- **KSGKT-S**: Similar to the KSGKT-Q embedding, but here we inputted the one-hot skill embedding to our KSGKT model.
- **KSGKT-Q&S**: Similar to the KSGKT-Q embedding, but here we concatenated the question and skill embedding and inputted the result to our KSGKT model.
- **KSGKT-CQE**: This model was the proposed model, renamed to emphasize that we learned the convolutional question embedding from the graph and inputted the result to our KSGKT model.

The AUC performances of the different question-embedding methods in the DKT, GIKT, and the proposed KSGKT models are displayed in Table 5.6. Herein, we mainly compared the one-hot embedding, the GNN-based graph embedding in GIKT, and the KS-enhanced graph embedding in the proposed KSGKT model. The one-hot embeddings in the DKT-extended models (DKT, DKT-Q, DKT-Q&S) and the KSGKT-extended models (KSGKT-Q, KSGKT-S, and KSGKT-Q&S) were outperformed by the other two embedding methods, validating that the embeddings learned from the graph incorporate more meaningful information about the graph (i.e., the relations among the questions and skills). Moreover, the KSGKT-extended models usually outperformed the corresponding DKT-extended models because they employ an attention mechanism. Comparing the performances of the DKT-CQE and DKT-extended models (also the KSGKT-CQE and the KSGKT-extended models), we validated the effectiveness of the KS-enhanced graph embedding method and its ability to improve the existing models. Comparing the performances of the GIKT and KSGKT-CQE models, we further confirmed that embedding learning on our KS-enhanced graph outperformed embedding learning on the original question–skill graph.

5.4.3 Visualization of Question Embedding Learned on a Graph

Figure 5.5 illustrates the question embeddings learned from the original and KS-enhanced graphs of Assist0910 and EdNet. For visual clarity, we randomly selected 50 skills and their corresponding questions in the two datasets and scattered the embedding vectors of these questions in the embedding matrix learned using Metapath2Vec. Specifically, we visualized the high-dimensional data by projecting the high-dimensional embedding vectors of the questions onto two-dimensional points. The projection was implemented using t-SNE [159] in Python.

As an example, we enhanced the original question-skill graph using the adjusted Kappa in Assist0910. As shown in Figure 5.5, the question embeddings of both datasets were highly structured: questions requiring the same skill tended to be clustered while questions belonging to different categories were well separated. Moreover, the question embeddings of the original graphs of both datasets were well separated into different categories, unlike the distributions in the KS-enhanced graph. This result is consistent with our motivation, as the KS is intended to bring the skill domain into the enhanced graph. Therefore, as each question is potentially associated with more skills (the associated skills plus the prerequisite skills), the distances between questions in

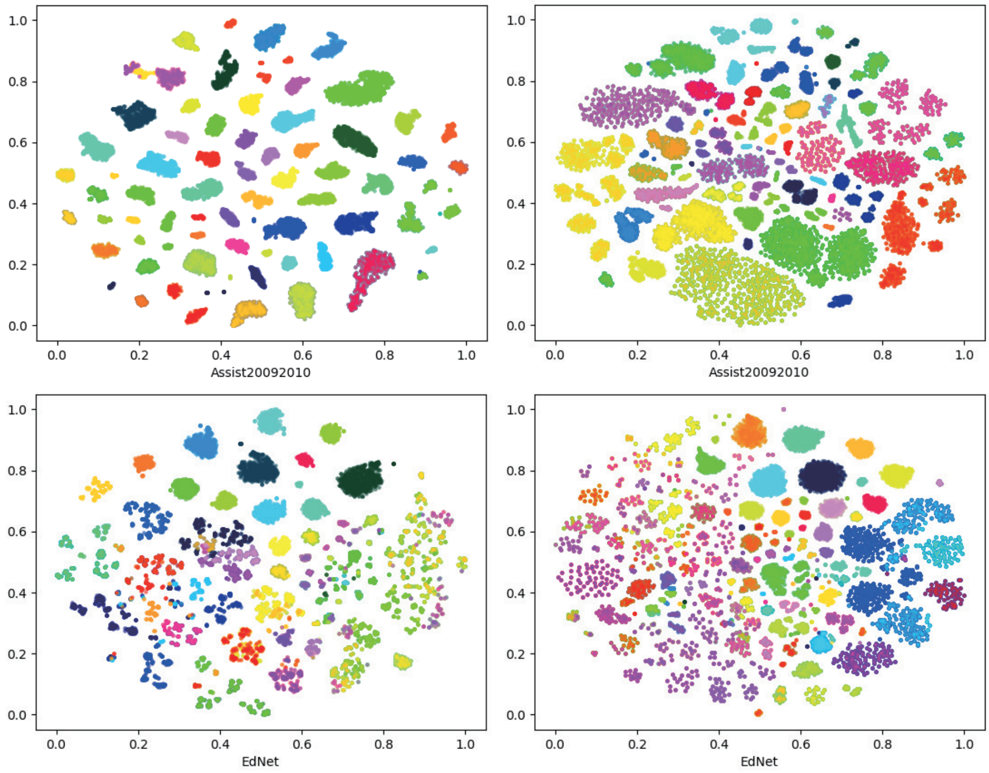
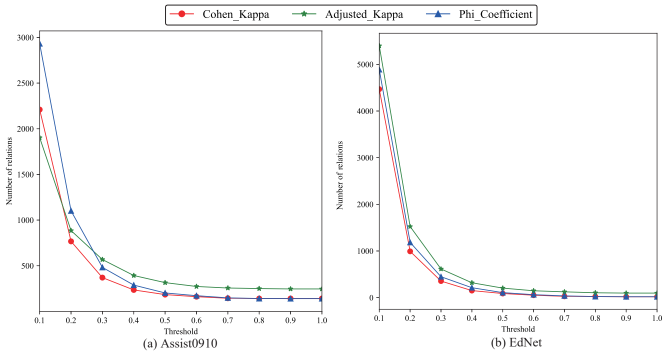


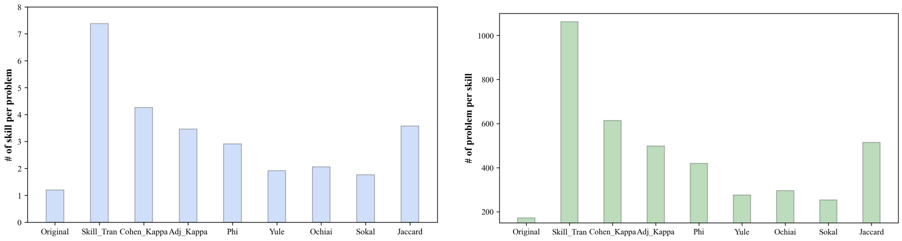
Figure 5.5: Embeddings of question nodes learned from the graphs of the Assist0910 and EdNet datasets. Each panel presents the questions associated with 50 kinds of skills in different categories (labeled with different colors). The upper left and right panels show the question embeddings learned from the original question–skill graph and the KS-enhanced graph using the adjusted Kappa in Assist0910, respectively. The lower figures show the question embeddings learned from the corresponding graphs in EdNet.

related categories became much closer. Also notable are the different distributions of the question embeddings learned from the original graphs of the Assist0910 and EdNet datasets. The boundaries of the questions requiring different skills were much clearer in the Assist0910 dataset than in EdNet. This result can be traced to the lower average number of skills related to each question in Assist0910 (1.207) than in EdNet (2.276). Therefore, a question in EdNet more likely requires multiple skills so the inter-group distances were much closer in the question embeddings of this dataset.



(a) Assist0910 (b) EdNet

Figure 5.6: Number of inferred relations in KS using the three methods with different thresholds on two datasets



(a) number of skills per question (b) number of questions per skill

Figure 5.7: Average number of skills per question and average number of questions per skill before and after considering the KS on the Assist0910 dataset

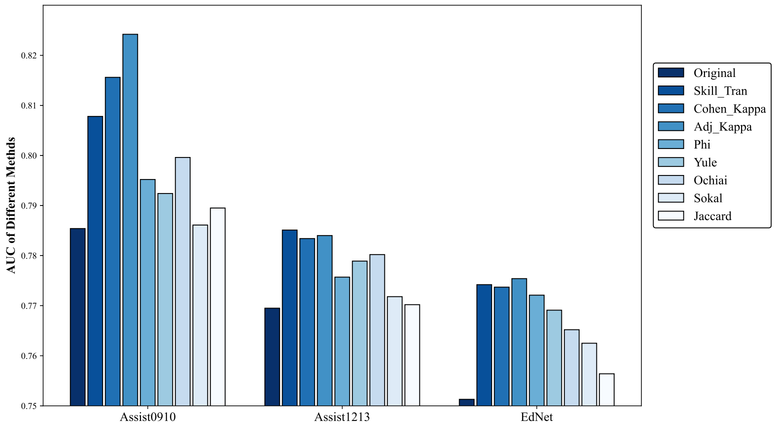


Figure 5.8: KS inferred from the learner response data

5.4.4 Comparison of KS Inferences by Different Methods

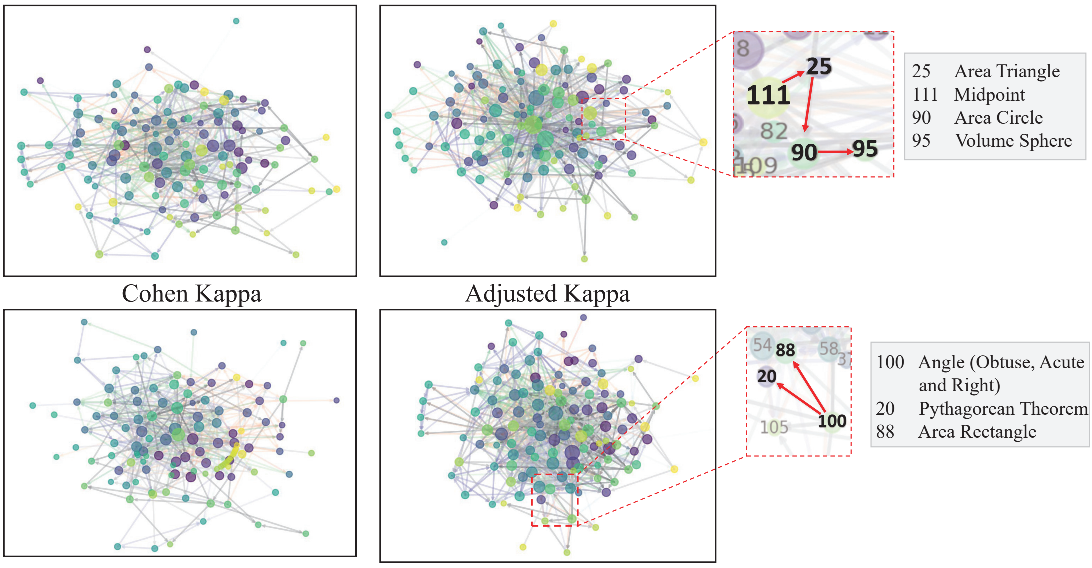
In subsection 5.2.1, we presented eight methods for inferring KS from the response data. Here we compares the performances of these methods.

The number of relations in the inferred KS can be controlled by a threshold. Here, the threshold for each method on the three datasets was decided on a case-by-case basis because the different coefficients have different ranges. Figure 5.6 shows the numbers of relations in KS with different thresholds after applying the Cohen Kappa, adjusted Kappa, and Phi coefficient on the Assist0910 and EdNet datasets. In our experiments, we set the threshold of both datasets to 0.2 to obtain a proper number of relations for the KS. Figure 5.7 compares the average number of skills per question and the average number of questions per skill before and after considering the KS on Assist0910. The original numbers of skills per question and questions per skill were both very low, but after incorporating KS with the questions using various methods, both values increased by different degrees, indicating that more meaningful information was included in the question representation.

Figure 5.8 compares the AUC results of the predictions of nine methods on the three datasets. Here the "original" method represents the embedding learning on the original question–skill relation graph and the other eight methods are based on the KS-enhanced graph. As evidenced in the figure, the eight KS enhanced methods generally outperformed the "original" method on all three datasets, validating the effectiveness of the KS enhanced graph in KT tasks. The adjusted Kappa yielded the best performance on both Assist0910 and EdNet, whereas the skill transaction method performed best on Assist1213.

5.4.5 Visualization of Inferred KS

Figure 5.9 illustrates the KS graphs of four methods, inferred from the learner response data in Assist0910. The right-hand side of this figure enlarges a part of the graphs to show their local connections. The nodes and edges in the KS form dense graphs with similar structures, showing several interconnected nodes. These graphs also show some interesting properties. In the adjust-Kappa graph, four nodes (25, 111, 90, and 95) were locally interconnected and revealed a perfect ordering of the skills (prerequisite and post-requisite relations) in the geometry. The local connections in the



Cohen Kappa

Adjusted Kappa

Skill Transition

Phi Coefficient

Figure 5.9: Visualization of the directed KS graphs generated by four methods on Assist0910

Phi coefficient graph also presented reasonable relations among the three skills. These results confirm that our KS discovery methods can infer prerequisite skill pairs from the ordering of learners' mastery of skills.

5.4.6 Effect of Convolutional Question Representation

To evaluate the effectiveness of the convolutional question-representation module, we fed the following question representations into our KSGKT model:

- **KSGKT with only QE**: The question embedding learned from the graph was input to our KSGKT model.
- **KSGKT with only SE**: The skill embedding learned from the graph was input to our KSGKT model.
- **KSGKT with QE&SE**: The question and skill embeddings learned from the graph were concatenated and input to our KSGKT model.
- **KSGKT with CQE**: This was the proposed model, renamed to emphasize that the convolutional question embedding learned from the graph was input to the KSGKT.

Table 5.7: Comparisons of AUC results of various KSGKT extended models with different question representations on the three datasets. The best results are marked with *.

Methods	ASSIST09	ASSIST12	EdNet
KSGKT with only QE	0.8005	0.7636	0.7489
KSGKT with only SE	0.8047	0.7569	0.7543
KSGKT with QE&SE	0.8132	0.7764	0.7603
KSGKT with CQE	0.8242*	0.7851*	0.7754*

Table 5.7 compares the performances of the KSGKT extended models. The KSGKT with the convolutional question embedding achieved the best results on all three datasets, indicating the effectiveness of the convolutional question-representation module. The KSGKT with QE&SE obtained the second best results, and the KSGKT with SE alone generally outperformed the KSGKT with QE alone (the exception was the Assist1213 dataset). The convolutional question representation acquires the distinctive features of the questions and also their interactions, thus improving the performance of the proposed model.

5.4.7 Effects of the Three Attention-calculating Methods

Table 5.8 compares the performances of the three attention methods on the datasets. The proposed method with the combined attention method far outperformed the proposed method with the shared skill-based and question similarity-based methods, indicating that incorporating the knowledge states of many related questions improved the model performance. However, the models with all three attention-calculating models significantly outperformed the model without attention, confirming that the attention mechanism benefits the KT task. This result is consistent with existing work [126, 127, 1], which reported that learners' past experience on related questions affects the performance on the current question.

Table 5.8: Comparison of AUC results of the proposed model with three attention-calculating methods on the three datasets. Best results are marked with *.

Methods	ASSIST0910	ASSIST1213	EdNet
Without attention	0.7998	0.7686	0.7523
shared skill-based attention	0.8229	0.7784	0.7729
question similarity-based attention	0.8233	0.7839	0.7727
combined attention	0.8242*	0.7851*	0.7754*

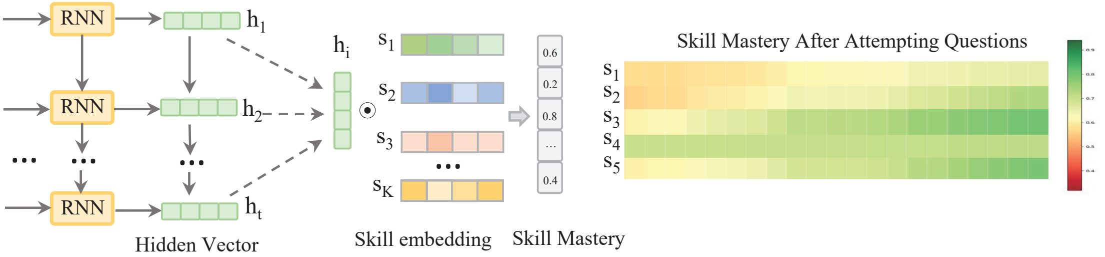


Figure 5.10: Obtaining a learner' knowledge state of each skill through interaction between the hidden knowledge states in the network and skill representation. The heat map on the right shows the evolution of the learner's skill mastery before and after attempting a series of questions. Each cell in the map indicates the learner's mastery of a specific skill at some time point. Over time, the mastery level increases from low (orange) to high (green) as the learner gradually masters the skills.

5.5 Case study

Most of the existing tutoring systems provide learners only with coarse-grained information such as correct/incorrect feedback or the score/rank of their exercise process [51, 33]. From a tutoring viewpoint, learners who understand the strengths and weaknesses of their knowledge points can remedy these weaknesses and improve themselves through self-regulated learning. From a teaching viewpoint, a comprehensive diagnostic report would help teachers identify the knowledge levels of both the whole class and individual students. Based on this information, they can design and provide timely interventions of the learning procedures.

Our model can easily generate a diagnostic report and provide the dynamic evolution of learners' mastery of each skill over time. Figure 5.10 shows the procedure of obtaining the dynamic skill mastery of learners. When a learner attempts a series of questions, the hidden layer of the network retains the relevant information in the learners' exercising history. Intuitively, this knowledge retention can be considered as an embedding of the general knowledge states of the learner [10] (shown as h_i). As the hidden vector and the skill embedding representation have the same dimension, we can map the hidden knowledge states vector and the skill embedding vector to the same space. The mastery level of each skill is then computed as

$$m(s_k) = \sigma(h_i \odot s_k) \in [0, 1] \tag{5.17}$$

where $\sigma(u) = 1/(1+exp(-u))$ is the sigmoid function and $\odot$ refers to the inner product. The right-hand side of Figure 5.10 illustrates the mastery levels of a learner on five skills in the Assist0910 dataset. Note that the skill mastery steadily evolved as the learner gradually mastered all five skills after attempting 21 questions. Assisted by this fine-grained diagnostic report, learners can focus on their weak knowledge without repeated training on their already mastered skills. This enlightenment will greatly improve students' learning efficiency.

Moreover, adaptive services such as remedial learning materials that meet learners' individual needs can be automatically provided based on the obtained knowledge mastery [12, 51]. Besides obtaining the required and prerequisite skills for solving a problem, our question-representation method infers the cognitive difficulty of the question from a learner's exercising history. Content predicted to be incompatible with the learners' knowledge level (too easy or too difficult) can be skipped or delayed, thereby effectively improving the students' learning paces while maintaining their engagement [12]. Many tutoring systems use similar exercises to identify whether a learner has mastered a certain type of exercise [51]. Our proposed method can be easily implemented for this purpose. In the question embeddings of the question–skill graph, questions linked by related skills are much closer in the embedding space than questions requiring different skills; hence, similar questions can be filtered from the question bank by performing simple similarity calculations on their embedding representations.

5.6 Summary

This chapter presented a knowledge structure-enhanced graph representation learning model for knowledge tracing (KSGKT) with an attention mechanism. By incorporating the knowledge structure into the knowledge tracing model, the model dynamically traces the learner's knowledge proficiency, thus alleviating the sparseness of the interaction data and the neglect of distinctive information related to the questions themselves and their relations (**Issue 2** in this thesis). These problems severely limit the efficacies of previous skill-based models.

To automatically obtain the KS in the domain, we first explored the abilities of eight methods to infer the domain KS from learner response data based on the mastery orders of pairs of skills. After integrating the KS with the KT procedure, we leveraged a graph-representation learning model and obtained the question and skill embeddings from the KS enhanced graph. To incorporate more distinctive information regarding the questions, we proposed a convolutional representation method that fuses the cognitive question difficulty with the question itself and its associated skill embedding. We thus obtained a comprehensive representation of each question. Feeding these representations into the proposed KT model with an attention mechanism, we can predict the learning performance on new problems. Extensive experiments conducted from six perspectives on three real-world datasets demonstrated the superiority of our model for learner performance modeling and KS discovery, validating its potential applicability to real educational environments.

Deep learning models, such as the proposed one in this chapter, have obtained excellent results to model the learning process by leveraging the powerful representation ability of the deep neural networks in a data-driven manner. However, deep neural network is regarded as a black-box, and most of the deep learning models retain the learner knowledge state in one hidden vector or as model parameters [119, 129, 32]. This works well for the prediction of learners' future performance, but from the perspective of proving good tutoring services to learners, their fine-grained knowledge proficiencies in a multi-granularity manner are particularly important. Note that although we can obtain the learner knowledge state on each skill using Eq. 5.17 in this chapter, this hidden vector representation of learners' general knowledge mixes the knowledge states on all the skills together, making it not precise enough to address the credit and blame assignment issue [40] for handling multiple KCs. Moreover, these methods found it difficult to go deeper into the explanation of the learners' performances in terms of their current knowledge proficiencies and item characteristics. Hence how to track and explain learners' evolving knowledge states simultaneously remains to be an important issue, which will be explored in the next chapter.

6
Knowledge Interaction Enhanced Sequential Modeling for Interpretable Learner Knowledge Assessment

As we have discussed in Section 2.3, the task of learner assessment is to obtain learner knowledge states based on the learners' explicit exercising logs. Based on the different application contexts, this task is addressed by two categories of educational psychology models: cognitive diagnostic assessment (CDA) models for *static testing* and knowledge tracing (KT) models for *dynamic learning*. For the testing context, CDA is to obtain the fine-grained diagnostic reports on learner knowledge instead of just the ranks or final scores. The data for analysis is the learners' performance data on a single summative quiz/test with limited items. For the learning context, KT is to obtain the learners' long-term evolving knowledge states for the purpose of providing adaptive

The material in this chapter is based on [160, 13] tutoring. The input data is generally the learners' long-term exercising logs in the systems.

CDA models have good interpretability because of the rich background educational psychology theories, but they are generally designed for the static assessment, and are difficult to meet the requirements of large-scale assessment. KT models dynamically track the evolution of learner knowledge, but most of the high-performance KT models are based on the deep neural networks and find it difficult to explain the results. Hence how to track and explain learners' evolving knowledge states simultaneously remains to be an important issue, this is the **Issue 3** in this thesis and will be explored in this chapter.

To alleviate this issue, this chapter proposes a novel model, called the *knowledge interaction-enhanced dynamic CDA* (KIEDCDA), to develop learner performance, and hence, dynamically diagnose and trace the evolution of each learner's knowledge proficiency during the exercise activities. Section 6.1 introduces the motivation of our proposed solution. Section 6.2 overviews the proposed solution in this chapter. Then the proposed KIEDCDA model is detailed in Section 6.3. Section 6.4 explains the experimental settings, and Section 6.5 presents the experimental results and analysis. Finally, a summary of this chapter is given in Section 6.6.

6.1 Motivation

Online learning systems have become increasingly intelligent in recent years with the application of techniques from artificial intelligence and cognitive psychology [14, 161, 162]. These systems generally model learner performance to assess their latent knowledge states, based on which many further adaptive services are provided to optimize learner learning. For example, tailored learning activities and support can be provided to meet individual learning needs and fulfil the diverse capabilities of learners. Moreover, a timely intervention of learning procedures by designing new measures and learning materials to remedy the weakness of learners can help teachers and administrators.

In the literature, massive effort has been devoted to both psychometrics and educational data mining, to propose various CDA models that can extract diagnostic information in a data-driven manner, such as the item response theory (IRT) [56], deterministic inputs, noisy "and" gate model (DINA) [62], multidimensional IRT (MIRT) [111], and matrix factorization (MF) [66]. These models have achieved great success in student assessment [114]. However, in practical scenarios, a good rule for setting the set of KCs for these models, except for IRT-based models, is that the number of KCs must not be too large (generally less than 10), so as to be statistically supportable [112, 113], which makes the assessments of a large number of KCs impractical [114], especially in large-scale adaptive learning environments. Conversely, IRT-based assessments provide coarse-grained uni- or low-dimensional values to represent the general proficiency of learners, which may not directly represent their strengths and weaknesses. Despite this limitation, IRT-based models have been widely used in practical assessment because of their interpretability to explain the learner performance in terms of the current knowledge proficiency and item characteristics. These CDA models are also generally used in independent assessments at some time points (i.e., performing CDA from a static perspective). However, learners' knowledge construction process is not static, but evolves over time, because learners learn and forget over time [9], as has long been converged by educational psychologists [29, 2].

Accordingly, several studies in the field of educational data mining have been conducted to dynamically track the evolution of learners' knowledge proficiency by considering their long-term exercising procedures. The pioneer model, called deep knowledge tracing (DKT) is the most popular model [10], which captures complex representations of student knowledge using the hidden variables of recurrent neural networks (RNNs). Although it obtains substantial improvement in terms of model performance, the hidden vector of learners' general knowledge mixes the knowledge states on all the KCs together, making it difficult to explain the mastery degrees of learners on each specific KC. To overcome this issue, a new neural network-based model, called the dynamic key-value memory network (DKVMN), was proposed [32] based on the memory-augumented neural network (MANN) [163, 164], which models students' knowledge states over all underlying KCs separately using an auxiliary key-value memory. This model shows a good representation capability to track and present the evolution of knowledge proficiencies on the underlying KCs. However, DKVMN retains the knowledge acquisition from the exercising sequences into the network parameters; thus, the causality between the learner performance and knowledge proficiencies is difficult to explain. Therefore, this chapter proposes a dynamic CDA model that incorporates not only the ability to trace the evolution of learners' knowledge proficiencies over time for large-scale assessments such as DKVMN, but also the interpretability to explain learner performance in terms of their current knowledge proficiency and item characteristics (e.g., IRT).

Meanwhile, previous experiences in exercise solving can affect the future ones because of the interdependencies between the KCs in these items (i.e., the prerequisites between pedagogical concepts). However, exploring the modeling of the skill interaction (specifying the interaction among KCs) for learner knowledge assessment is considerably underexplored because most of the CDA and KT models simply assume that all items and KCs are independent of each other. In the real world, when learners acquire knowledge growth from a certain item incorporating skill KC_1, they also improve the attainment of skill KC_2 to some extent, which we refer to herein as *knowledge interaction*. In the Chapter 5, we have explored to automatically discover the knowledge interaction of skills from learner response data by ordering the learners' mastery of skills. However, this method is separate from the KT procedure and may obtain imprecise results during a cold start. To cope with this, in this chapter we integrate the processes of knowledge interaction modeling and learner assessment together and optimize them simultaneously using the deep learning framework.

This chapter proposes a new dynamic CDA model, called the KIEDCDA model, which provides four advantages over the existing models.

- The KIEDCDA model unifies the strengths of the memory capacity of the key-value memory network to enhance the representation of the knowledge states during exercise solving, as well as the ability to trace the evolution of learners' knowledge proficiencies over time for large-scale assessments, and the interpretability of IRT to explain learner performance in terms of their current knowledge proficiency and item characteristics.

- The KIEDCDA model exploits the interdependencies between the KCs from the knowledge structure to incorporate the knowledge interaction into the CDA procedure. In modeling the learner learning, it automatically learns the interdependencies between the KCs from the learners' exercising logs. This knowledge interaction procedure not only improves the precision to infer the dynamic knowledge proficiencies of learners but also enhances the ability to

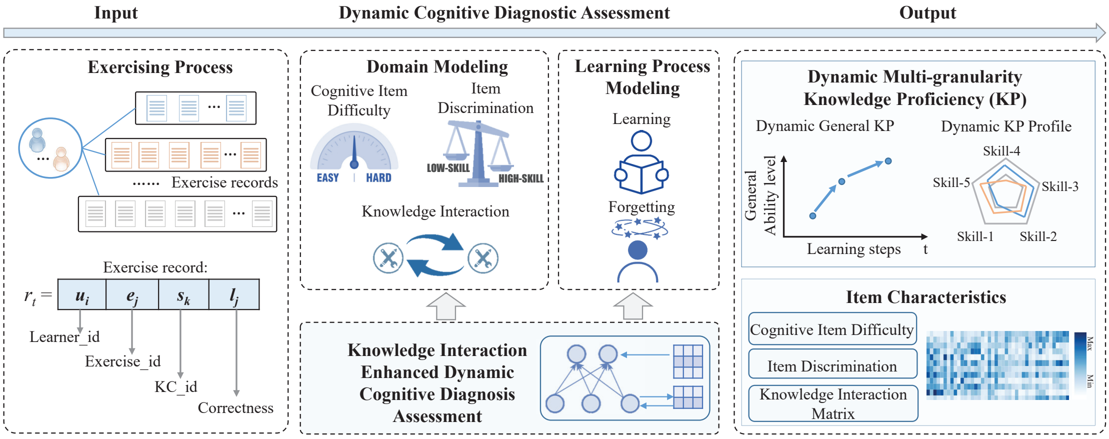


Figure 6.1: Framework of the proposed dynamic CDA.

capture the long-term dependencies in the exercise sequence.

- The KIEDCDA model can not only output the learners' knowledge proficiencies in a multi-granularity manner but also output the item characteristics to better model the learner performance.

Comprehensive experimental evaluations from six perspectives on five real-world datasets are conducted to test the proposed model. The results demonstrate the superiority and interpretability of our method in dynamically modeling the learning performance.

6.2 Solution Overview

The framework of the proposed KIEDCDA model is shown in Figure 6.1. After inputting the learners' exercise records, the KIEDCDA model begins to perform the learner performance modeling on these records from timestamp t_1 to the latest timestamp. Specifically, the model conducts a diagnosis assessment from two inter-connected aspects: domain modeling and learning process modeling. Domain modeling studies the factors within a domain of items that may affect the leaner performance, and learning process modeling considers a learner's real learning process. Our model establishes these two aspects because a learner's performance on items is highly related to his/her previous learning process, as well as the items he/she has interacted with. We perform domain modeling from three aspects: cognitive item difficulty, item discrimination, and knowledge interaction. The former two are characteristics related to each item (as considered in IRT). The knowledge interaction models the interaction effect between different KCs in the defined domain. Moreover, the learning process is modeled by considering the learners' dynamic learning and forgetting procedures. The evolution of the learners' knowledge proficiencies and the item characteristics can be obtained after performing the dynamic CDA. Coarse- and fine-granularity proficiencies can both be obtained, providing a more delicate proof for further adaptive services. Furthermore, the cognitive item difficulty and discrimination for each learner on each item at the current timestamp are obtained, making it possible to delve deep into the causality between the learner performance and his/her proficiency and item characteristics. The knowledge interaction matrix can also be automatically learned from the input data, which can be potentially used to analyze the structure of KCs within a defined domain.

6.3 KIEDCDA

This section presents the proposed KIEDCDA model. First, we introduce the KIEDCDA model architecture and then show how previous learning procedures are encoded into the network and utilized to model the learner performance.

6.3.1 Model Architecture

Figure 6.2 shows the KIEDCDA model architecture augmented by a key-value memory following [32], to trace the evolution of learners' knowledge proficiencies over time for a large-scale assessment, and the IRT framework to enable the interpretability of explaining the learners' performances in terms of their current knowledge proficiencies and item characteristics. The key-value memory $\langle M^K, M_t^V \rangle$ (green part, Figure 6.2) is a pair of external memory for storing paired parameters in the model. M^K with size $N \times d_k$ is a static matrix for storing all embedded N underlying KCs with dimension d_k. M_t^V with size $N \times d_v$ is a dynamic matrix for recording a learner's knowledge proficiency of the corresponding KC at timestamp t (i.e., M_t^V is different from M_{t+1}^V because a learner's proficiencies evolve over time). Every time a learner attempts an

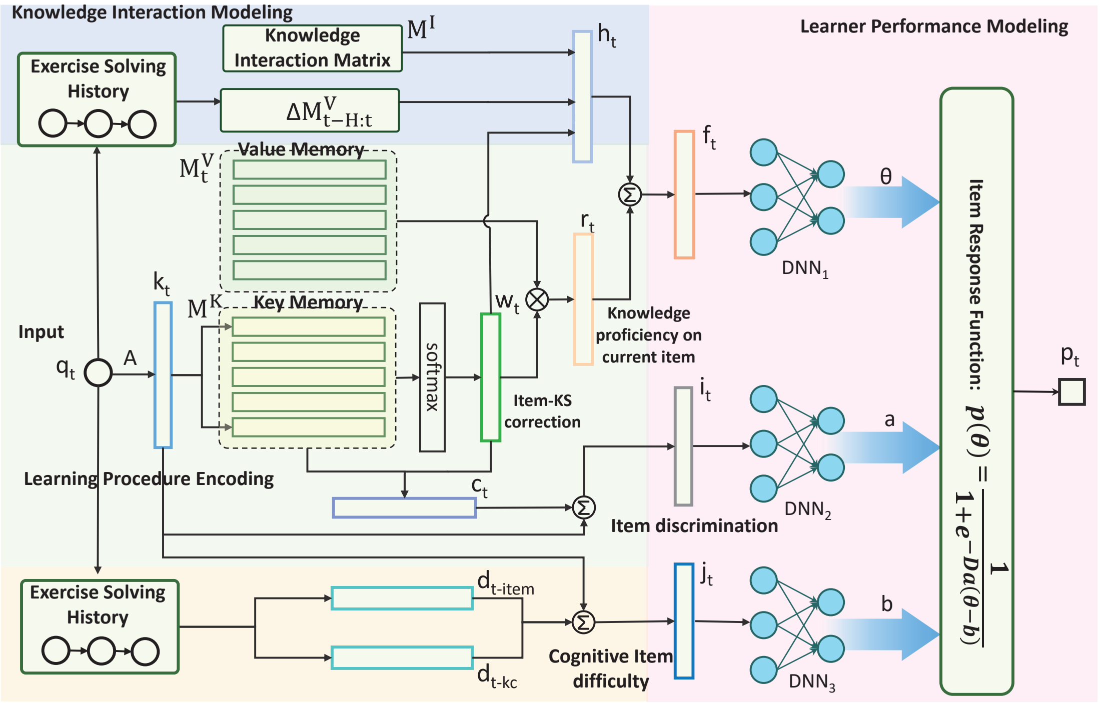


Figure 6.2: Architecture of the proposed KIEDCDA model. Previous learning procedures are encoded into the network to obtain three parameters (i.e., θ, a, and b) representing learner general proficiency, item discrimination, and item difficulty, and are input into the IRT to model the learner performance. $\bigotimes$ and Σ⃝ represent element-wise multiplication and concatenation of vectors, respectively.

item with some underlying KCs, the corresponding space in M_t^V is updated based on the proficiency change through the currently attempted item. Therefore, the key-value memory acquires a proficiency change based only on the most recent item and fails to capture the impact of knowledge interaction of the previously attempted exercises on the current one (i.e., the long-term dependencies in the exercising sequence). To cope with this, we incorporate a knowledge interaction matrix M^I (blue part, Figure 6.2) to model the interactions between each pair of KCs. This process considers the influence of the knowledge proficiency growth $\Delta M_{t-H:t}^V$ by attempting the latest H previous items in the exercising sequence to the KCs in the current item, and thus, further improving precision in inferring current learner proficiencies. Therefore, we use the proficiencies in M_t^V and the knowledge interaction procedure to infer the learners' real ability. To accurately model the item difficulty to the performance of learners, we consider the previous items and KCs trained in the learners' exercise-solving history to model the learner-oriented cognitive item difficulty for the current item (orange part, Figure 6.2). Moreover, the item discrimination for each item is modeled based on the KCs contained in each item. Finally, the IRT framework is used to explain the learner performance (p_t) in terms of their proficiencies (M_t^V and θ) and item characteristics (a and b) (pink part, Figure 6.2).

In the subsequent subsections, we will specify how the KIEDCDA model worked to perform a dynamic CDA.

6.3.2 Input

The models take the current item q_t and the previously attempted items in the exercising sequence as the input. Item q_t is first embedded into a vector k_t with dimension d_k by looking up an embedding matrix $A \in \mathbb{R}^{J \times d_k}$.

$$k_t = embedding(q_t, A) \tag{6.1}$$

Each of the previous items is labeled with the proficiency matrix M_t^V at a specific timestamp.

6.3.3 Learning Procedure Encoding

Given the input to our model, it must encode the interaction procedures between the learners and items to the network by conducting domain and learning process modeling. The relevant details are as follows:

Item–KC Correction: After obtaining the embedding vector k_t of item q_t, our model will query the KC embedding matrix M^K to find the correction between item q_t and the underlying KCs in M^K. A correction vector w_t can be obtained by computing the softmax of the inner product between k_t and all slots in M^K, where the element $w_t(i)$ represents the degree of dependence on the specific KC to solve the item.

$$w_t(i) = softmax(k_t^T M^K(i)) \tag{6.2}$$

Knowledge Interaction: The knowledge interaction captures the long-term dependencies in the exercising sequence to model the influence of a previous experience to the current item. Our model evaluates the knowledge interaction based on a knowledge interaction matrix M^I and the latest previous H items. M^I is a symmetric matrix, in which each element I_{ij} represents the impact of KC s_i on s_j. Note that the elements in the diagonal direction are set to 1, and I_{ij} and I_{ji} are always equal. H is a hyperparameter for modeling the influence of the previous H items to the current one. It calculates the influence of the proficiency changes by attempting these H items to the KCs in the current item. Note that every time a learner attempts an item, only the slots of the involved KCs in the proficiency matrix M_t^V are updated. Therefore we calculate the proficiency changes $\Delta M_{t-H:t}^V$ (in the following, we will use ΔM^V for convenience) from timestamp $t-H$ to t by subtracting M_{t-H}^V from M_t^V. The influence vector h^t is calculated as follows:

$$\Delta M^V = M_t^V - M_{t-H}^V \tag{6.3}$$

$$h^t = (\Delta M^V)^T M^I w_t = \sum_{i=1}^{N} \sum_{j=1}^{N} \Delta M^V(i) I_{ij} w_j \tag{6.4}$$

Knowledge Proficiency on Current Item: With the item–KC correction vector, learners' knowledge proficiencies on the current item can be retrieved from the proficiency matrix M_t^V. However, considering the knowledge interaction of the accumulated long-term experience to the current item, we calculate the learners' knowledge proficiencies herein by incorporating them together (i.e., the knowledge evolution is based on the knowledge growth from not only the latest item but also a previous experience). We calculate the KC-wise weighted sum of $M_t^V(i)$ and $w_t(i)$ as r_t to retrieve the proficiencies of related KCs from M_t^V, and concatenate it with the influence vector h_t of the previous experience to obtain the final knowledge proficiency vector f_t on the current item q_t.

$$r_t = \sum_{i=1}^{N} w_t(i) M_t^V(i) \tag{6.5}$$

$$f_t = concatenate[r_t, h_t] \tag{6.6}$$

Item Discrimination: The item discrimination describes how well an item can differentiate learners who have mastered the involved KCs from those who have not [56]. The discrimination highly depends on the item itself and the KCs involved in the item. Following [165], we calculate the item discrimination from the KCs corresponding to the item. We specifically use the item–KC correction as the weight to sum the KC embeddings in M^K and obtain a vector c_t with d_k dimension. We then concatenate it with the embedding vector k_t of item q_t to obtain the final item discrimination vector i_t.

$$c_t = \sum_{i=1}^{N} w_t(i) M^K(i) \tag{6.7}$$

$$i_t = concatenate[c_t, k_t] \tag{6.8}$$

Cognitive Item Difficulty: We use the cognitive item difficulty proposed in Section 4.3.1 to model the learner-oriented item difficulty in the dynamic learning process based on the learners' current cognition (i.e., their current cognitive structures built during the exercising solving procedure). The cognitive difficulty $\Psi_{item,t}$ and $\Psi_{kc,t}$ from the aspects of the previous same items and KCs are calculated using Eq. 4.2. Then we represent them as one-hot vectors of length $c + 1$. We can obtain the final cognitive difficulty vector of item q_t by concatenating them and the embedding vector k_t.

$$d_{t-item} = onehot(\Psi_{item,t}) \tag{6.9}$$

$$d_{t-kc} = onehot(\Psi_{kc,t}) \tag{6.10}$$

$$j_t = concatenate[d_{t-item}, d_{t-kc}, k_t] \tag{6.11}$$

Proficiency Update: After a learner attempts an item, our model will update his proficiencies in the proficiency matrix M_t^V based on the knowledge attainment from attempting this item. The updating rule aims to change the proficiency of the KCs involved in the current item whether learners attempt correctly or incorrectly on the current item. Figure 6.3 shows the updating process for changing M_t^V to M_{t+1}^V.

Following [123], our model considers the attainment from attempting the current

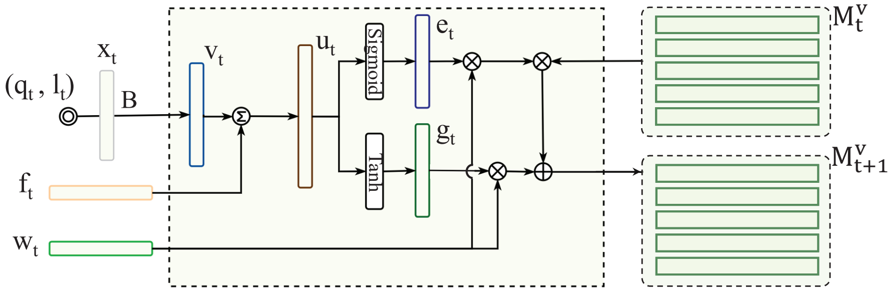


Figure 6.3: Updating process of the knowledge proficiency memory from M_t^V at timestamp t to M_{t+1}^V at timestamp $t+1$.

item and the learners' current knowledge proficiencies to update the proficiency matrix. The current item q_t and the correctness of the learner's answer l_t, as well as the learner's current knowledge proficiency are specifically input to the update module. We first represent q_t and l_t as a vector x_t and embed it by looking up an embedding matrix $B \in \mathbb{R}^{2J \times d_v}$ to obtain the final representation v_t for this learning log. The v_t and f_t concatenation is then input into this updating module.

$$x_t \in \{0,1\}^{2J} : \begin{cases} x_t^{q_t} = 1, & if\ l_t = 0 \\ x_t^{q_t+J} = 1, & if\ l_t = 1 \end{cases} \tag{6.12}$$

$$v_t = embedding(x_t, B) \tag{6.13}$$

$$u_t = concatenate[v_t, f_t] \tag{6.14}$$

Similar to general memory networks [32], the updating process incorporates two gates: erase gate and add gate. The erase gate controls the information to be erased from the proficiency matrix that captures the forgetting behavior of learners in the learning process. The add gate mimics the learning behavior of learners because it controls the information to be added into the proficiency matrix due to knowledge growth. Thus, an erase vector e_t and an add vector g_t can be obtained. The updating process for M_t^V to transit to M_{t+1}^V is described as follows:

$$e_t = sigmoid(W_e u_t + b_e) \tag{6.15}$$

$$g_t = tanh(W_g u_t + b_g) \tag{6.16}$$

$$M^V_{t+1}(i) = M^V_t(i) \otimes [1 - w_t(i)e_t] + w_t(i)g_t \tag{6.17}$$

6.3.4 Learner Performance Modeling

The learner performance can be modeled after encoding the learning procedure into the network. We have already obtained a learner's current knowledge proficiency vector f_t, the item discrimination vector i_t, and the cognitive item difficulty vector j_t on the current item. Thus, we now use the IRT framework to model the learner performance and obtain the explainable causality between the learner's proficiency and item characteristics.

Before inputting these components into IRT, deep neural networks are used to automatically learn high-order, nonlinear features from these vectors and transform them into meaningful values. Three DNNs (i.e., DNN_1, DNN_2, and DNN_3) are used herein to obtain three parameters in IRT: learner ability θ, item discrimination a, and item difficulty b. The learner performance on item q_t is modeled in Eq. (23) as p_t. Following the existing study [166, 56] on the requirements of item discrimination and difficulty, we normalize both to the [-4,4] range in Eqs. (21) and (22).

$$\theta = DNN_1(f_t) \tag{6.18}$$

$$a = 8 \times (sigmoid(DNN_2(i_t)) - 0.5) \tag{6.19}$$

$$b = 8 \times (sigmoid(DNN_3(j_t)) - 0.5) \tag{6.20}$$

$$p_t = \frac{1}{1 + e^{-1.7a(\theta - b)}} \tag{6.21}$$

6.3.5 Optimization

The cross-entropy loss was used to optimize our model by minimizing the following objective function between the true answer l_t and the predicted performance p_t on each item in the sequence.

$$\mathcal{L} = -\sum_{t \in [1:T]} (l_t log\, p_t + (1 - l_t) log(1 - p_t)) \tag{6.22}$$

All the model and network parameters are updated in each iteration by minimizing the above loss function using Adam optimization.

6.4 Experimental Settings

Several experiments are conducted to evaluate the proposed KIEDCDA model on five public real-world datasets for various tasks. This section describes the experimental settings, including the aims, datasets, comparison baselines, setup and implementation, and evaluation metrics. The detailed experimental results and model analysis are presented in the next section.

6.4.1 Experimental Aims

We conduct experiments to answer the following questions:

1. Based on the inferred knowledge proficiency from the exercise history, how does KIEDCDA perform on predicting a learner's performance on new items? (see Section 6.5.1)

2. What is the optimal set of hyperparameters for the neural network of KIEDCDA and their sensitivities? (see Section 6.5.2.A.)

3. Up to what degree are the contributions of each KIEDCDA component to the performance of the whole model? (see Section 6.5.3)

4. How does KIEDCDA perform on capturing the knowledge interaction between each latent KC? (see Section 6.5.2.B.)

Table 6.1: Comparison of the characteristics of our model and the baseline models.

Model	Model Character		Domain modeling		Knowledge Modeling		Dynamic Intepretation
	Knowledge Interaction	Concept Discovery	Item discrimination	Item difficulty	Evolution of KC proficiency	Score Prediction	
IRT	×	×	✓	constant	×	✓	✓
MIRT	×	×	✓	constant	✓	✓	✓
LFA	×	×	×	constant	×	✓	×
PFA	×	×	×	constant	×	✓	×
KTM	×	×	×	constant	×	✓	×
DKT	×	×	×	×	×	✓	×
DKVMN	×	✓	×	constant	✓	✓	×
KIEDCDA	✓	✓	✓	Adaptive	Multi-granularity	✓	✓

5. How does KIEDCDA perform on discovering the correlation between items and latent KCs? (see Section 6.5.2.C.)

6. How does KIEDCDA perform on knowledge and domain modeling (i.e., capturing the evolution of knowledge proficiency and item parameters) in the specific domain? (see Section 6.5.4)

6.4.2 Datasets and Baseline Models

Five well-established datasets were used to conduct the experiments, namely Algebra0506, Statics2011, Assist0910, Bridge2Algebra0607, and Assist1213, the detail of these datasets are described in Section 3.2. We compare the proposed KIEDCDA model with seven of the best-known state-of-the-art models (i.e., DKT, DKVMN, KTM, IRT, MIRT, PFA, and LFA) to demonstrate its effectiveness. These models are chosen because of their predominance in either educational psychometrics (i.e., IRT and MIRT) or educational data mining (i.e., DKT, DKVMN, KTM, PFA, and LFA); the first three are also the best performers in this field. DKT and DKVMN are neural network models. KTM is based on the factorization machine in the data mining field, while the other models are non-neural network models. The model details are elaborated below:

- *IRT:* IRT [56] is a basic model, which is the most popular CDA model. It is a regression model used to model the probability of a student answering an item correctly based on his/her ability and the item difficulty.

- *MIRT:* MIRT [111] extends IRT by considering the interactions of the multidimensional embedding vectors of the learner ability and the item difficulty.
- *LFA:* LFA [95] is a factor analysis model that models the probability of attempting an item correctly by considering the difficulty of the KCs involved in the item and the number of attempts on items requiring the involved KCs.
- *PFA:* PFA [96] improves the LFA by considering successful and failed attempts separately. Both PFA and LFA assume that learners share the same learning rate in their learning process.
- *KTM:* KTM [30] is a newly proposed model based on the factorization machine. It models the probability of exercising results (right or wrong) based on a sparse set of weights for all features in a sample. It is a generic framework that incorporates side information (e.g., users, items, skills, win and fail attempts) into the student model [29].
- *DKT:* DKT[10] is the first model to use a deep neural network for conducting CDA. It considers the prediction of learner knowledge as a sequence learning task and leverages recurrent neural networks to capture the complex representations of student knowledge using the hidden variables of RNNs. By learning from the input sequences of the students' learning history, the hidden layer retains relevant information that can be intuitively seen as embedding the knowledge states of learners [119].
- *DKVMN:* DKVMN [32] is a recent state-of-the-art deep learning-based model that establishes learners' knowledge states using an auxiliary memory to augment the neural networks. It embeds the skill information into a key matrix and accumulates temporal information from the learners' exercising sequences to infer their knowledge states on these skills. It then stores them into a value matrix. However, it considers the changing of the knowledge state only from the most recent item, and does not consider knowledge interaction among skills, as well as the cognitive item difficulty.

For a better illustration, we show the model characteristics in Table 6.1.

6.4.3 Setup and Implementation

Before conducting the experiments, we take 30% of the sequences in a dataset as the test set and the other 70% as the training set, and perform a five-fold cross-validation on the training set to find the optimal set of hyperparameters through grid search. We run the training and evaluation processes for five times during the test phase; hence, the mean and standard deviation of the experimental results are reported in the experimental analysis.

The item embedding matrix **A**, item-response embedding matrix **B**, key memory matrix M^K, value memory matrix M^V, and other model parameters (**W** and **b**) are randomly initialized from a Gaussian distribution with a zero mean and a standard deviation of 0.1. For the knowledge interaction matrix M^I, we use an initializer from the standard uniform distribution to keep the interaction coefficient of each KC pair between the range of 0 and 1. We only use the coefficients in the upper triangle and set the same value for the corresponding coefficients in the lower triangle to ensure M^I as a symmetric matrix. All parameters are learned in the training procedure by optimizing the model using cross-entropy loss.

We optimize the model using Adam optimization with the learning rate case-by-case in the five datasets, and consistently set the the norm clipping threshold to 10 and the batch size to 32. The sequence length to input into the model is fixed to 200. Thus, long sequences are divided into several short ones and short sequences are padded with a null symbol to maintain it at a length of 200.

We use the open source to implement KTM[1], and DKVMN[2]. The proposed model and DKT are implemented using TensorFlow. We conduct the IRT, MIRT, LFA and AFA experiments on all datasets based on the KTM framework. KTM is implemented using various features, including items, skills, wins, and fails. The size of the hidden layers for the DKT model is chosen from {10, 50, 100, 200}. Meanwhile, the size of the key memory matrix M^K for DKVMN and the proposed KIEDCDA model is chosen from {5, 10, 20, 50, 100}. The dimension of the key and the value memory slot are chosen from {10, 50, 100, 200}. For the proposed KIEDCDA model, we set the latest previous items considered in the knowledge interaction process for the current item from {1, 5, 10, 20, 50}. We simplify the grid search in quite a large search space by setting $d_k = d_v$, to make the dimension of the key and value memory slot similar. Our model is tested for 50 epochs because it is sufficient for the model convergence. For the other baselines, we follow the settings of the best parameters in the original work. All models are

[1] https://github.com/jilljenn/ktm

[2] https://github.com/jennyzhang0215/DKVMN

Table 6.2: Comparisons of the AUC results among different models on the five datasets.

Dataset	Test AUC (%)							
	KIEDCDA	DKT	DKVMN	KTM	IRT	MIRT	PFA	LFA
Algebra0506	**85.09 +/- 0.04**	82.94+/-0.20	84.21 +/- 0.17	72.79+/-1.05	77.11+/-0.41	77.21+/-0.93	75.38+/-0.95	72.15+/-0.46
Statics2011	**82.74+/-0.04**	80.46+/-0.35	**82.77+/-0.07**	80.31+/-0.56	78.61+/-0.33	78.44+/-0.35	68.40+/-0.99	65.40+/-0.37
Assist0910	**83.08 +/- 0.27**	80.97+/-0.12	81.49+-/0.06	73.82+/-0.40	67.70+/-0.13	67.69+/-0.55	69.98+/-0.70	62.09+/-0.47
Bridge2Algebra0607	**79.72 +/- 0.32**	77.55+/-0.43	79.03+/-0.06	77.43+/-0.86	74.76+/-0.44	74.70+/-0.23	74.53+/-0.32	70.67+/-0.62
Assist1213	**73.43 +/- 0.09**	72.12+/-0.10	72.66+/-0.11	**73.42+/-0.07**	70.18+/-0.25	70.13+/-0.19	66.89+/-0.21	60.99+/-0.13

trained and tested on the same training and testing sets.

To report the results, the widely used AUC, ACC and NLL are used as the evaluation metrics. The details of these metrics are described in Section 3.3.

6.5 Results and Analysis

This section presents the experimental results. We answer the first research question in Section 5.1 by comparing the learner score prediction performances of our proposed KIEDCDA model and all baseline models in Section 6.1. We conduct a model analysis from the three aspects in Section 6.2 to answer the second, fourth, and fifth questions. In Section 6.3, we conduct an ablation study to evaluate the contributions of each KIEDCDA component and answer the third question. Section 6.4 presents the KIEDCDA results on knowledge and domain modeling. We present its abilities for tracing the evolution of the multi-granularity knowledge proficiency and item parameters to answer the sixth question.

6.5.1 Learner Performance Prediction

We claim herein that the proposed KIEDCDA model can infer the current knowledge proficiency of learners based on their exercising history. Directly evaluating this estimated information is not easy because obtaining the actual knowledge proficiency in the human brain is difficult [2]. As an alternative, following the existing studies [56, 95, 96, 30, 32] , we evaluate herein the models by the performance of learner score prediction in the future based on the estimated current knowledge proficiency. We also compare our KIEDCDA model with all baseline models on five datasets. Table 6.2 presents the AUC results of all models. Figure 6.4 depicts the ACC and Loss results.

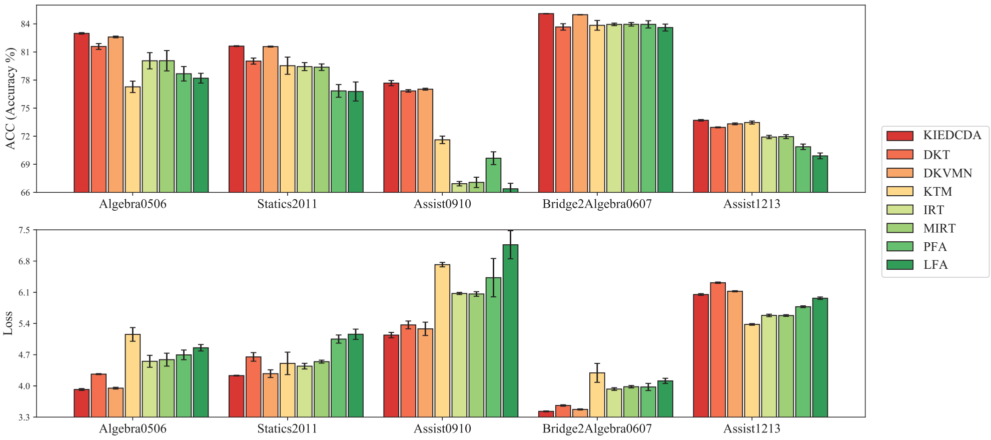


Figure 6.4: Comparisons of the ACC and Loss results among different models on the five datasets. The upper bar graph shows the comparisons of ACC. The lower graph illustrates the comparisons of Loss.

Our model outperforms the other models over all five datasets, except on the Statics2011 dataset, where it obtains a slightly smaller AUC compared to the DKVMN model. However, it has a much larger ACC and a smaller Loss than DKVMN on that dataset. In the Algebra0506 dataset, our KIEDCDA model achieves the best average test AUC of 85.09%; DKT obtains an AUC of 82.94%, and DKVMN ranks the second best at 84.21%. The other models only obtain an AUC of less than 78%. In the Statics2011, our model and DKVMN obtain very similar results, which is probably because Statics2011 has the smallest number of learners and total items and the fewest interaction entries between the learners and the items. Compared to the DKVMN model, this data volume may not be good enough for a more complex network structure of KIEDCDA in terms of optimizing more parameters; however, both models are better than the other models by at least 2% AUC. Our model outperforms all other models in the Assist0910, Bridge2Algebra0607 and Assist1213 datasets. In the Assist1213, KTM is the second-best performer with an AUC of 73.42%, which is slightly smaller than that of the KIEDCDA model, but is still better than DKVMN and DKT. The overall performance of all models on Assist1213 is the lowest among all datasets, indicating the difficulty of performance prediction on this dataset caused by the large numbers of learners and the small number of average attempted items per learner, which makes the training process difficult considering the limited sequence information that can be utilized. The ACC comparisons in Figure 6.4 also show that our KIEDCDA model outperforms all other models in all five datasets. The Loss comparisons also illustrate a similar property. Furthermore, the neural network-based methods present better performances than the non-neural-network models on this task.

In summary, KIEDCDA shows the best performance for the learner score prediction task compared to all state-of-the-art models while achieving the second-best AUC among all models on the Statics2011 dataset; however, note that the results it obtained are very close to the best results obtained by DKVMN. DKVMN generally has the second-best performance except on the Assist1213 dataset. These results demonstrate that our KIEDCDA can utilize the knowledge interaction of the previous exercising history on the current item and the cognitive item difficulty to better estimate the learners' current knowledge proficiency and predict their future performance based on this information.

6.5.2 Model Analysis

In this subsection, we further deeply analyze our proposed model on three aspects: parameter sensitivity, the ability to model the knowledge interaction between the latent KCs, and ability to discover the underlying KCs for each item.

A. Sensitivity of the Model Parameters

We will now discuss the parameter sensitivity in the proposed KIEDCDA model. Specifically, three hyperparameters are crucial for the model performance: the number of slots in the key memory matrix ($m.size$), which represents the number of latent skills; the dimension of each slot in the key memory matrix ($s.dim$), which indicates the dimension of skill representation; and the number of latest previous exercising records (i.e., related items), considered in the knowledge interaction procedure ($ri.num$). We consider different sets of these three parameters and analyze their influence on the AUC results of the learner performance prediction.

Table 6.3 shows the results of different parameter sets on the five datasets. We report $m.size = \{5, 10, 20, 50\}$ for the Statics2011 dataset because when the m.size is larger than 50, it does not result in any improvement. We report $m.size = \{10, 20, 50, 100\}$ for the other four datasets for the same reason of the m.size being smaller than 10. Fixed on each $m.size$, we set $s.dim = \{10, 50, 100, 200\}$ and $ri.num = \{1, 5, 10, 20, 50\}$ and test

Table 6.3: Comparisons of AUC results using different hyperparameter sets.

Algebra0506				Statics2011				Assist0910				Bridge2Algebra0506				Assist1213			
m.size	s.dim	ri.num	AUC	m.size	s.dim	ri.num	AUC	m.size	s.dim	ri.num	AUC	m.size	s.dim	ri.num	AUC	m.size	s.dim	ri.num	AUC
10	10	5	84.15	5	10	1	82.61	10	10	1	82.91	10	10	10	78.68	10	10	1	72.82
10	50	5	84.48	***5***	***50***	***20***	***82.74***	10	50	10	82.89	10	50	1	79.13	10	50	5	72.91
10	100	20	84.77	5	100	10	82.57	10	100	50	82.81	10	100	5	79.32	10	100	5	73.05
10	200	5	84.89	5	200	20	82.72	10	200	10	82.33	10	200	1	79.39	10	200	10	73.01
20	10	10	83.73	10	10	50	82.63	20	10	10	82.92	20	10	10	79.01	20	10	5	72.92
20	50	50	84.35	10	50	20	82.71	20	50	1	82.80	20	50	5	79.17	20	50	5	73.00
20	100	1	84.95	10	100	5	82.32	20	100	50	82.68	20	100	5	79.38	20	100	50	73.11
20	200	5	85.03	10	200	1	82.69	20	200	50	81.91	20	200	1	79.39	20	200	20	73.14
50	10	5	83.72	20	10	1	82.54	***50***	***10***	***1***	***83.08***	50	10	20	78.91	50	10	10	73.07
50	50	5	84.42	20	50	5	82.56	50	50	1	83.01	50	50	20	79.03	50	50	5	73.18
50	100	1	84.68	20	100	10	82.63	50	100	50	82.59	50	100	1	79.36	***50***	***100***	***5***	***73.42***
50	200	5	85.02	20	200	1	82.23	50	200	5	82.52	***50***	***200***	***10***	***79.72***	50	200	50	73.04
100	10	1	83.84	50	10	5	82.11	100	10	10	82.71	100	10	10	78.94	100	10	20	73.05
100	50	5	84.22	50	50	5	82.65	100	50	1	82.89	100	50	5	78.87	100	50	10	73.02
100	100	10	84.69	50	100	50	82.49	100	100	20	82.61	100	100	1	79.32	100	100	5	73.17
100	***200***	***10***	***85.09***	50	200	1	82.41	100	200	20	82.77	100	200	5	79.14	100	200	1	73.10

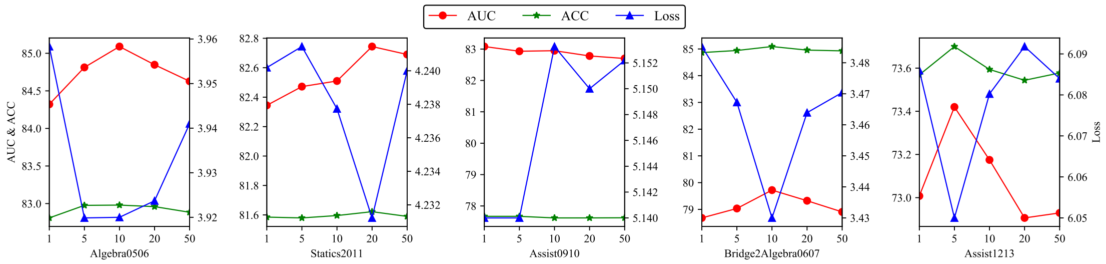


Figure 6.5: Impact of different *ri.num* on the five datasets.

the combinations for all datasets. We only report herein the best AUC results on each *m.size* and *s.dim* combination and on a specific value in *ri.num*.

In the Algebra0506 dataset, the AUC results show an increasing trend with the increasing *s.dim* on each fixed *m.size* and achieve the best result with *m.size* = 100, *s.dim* = 200, and *ri.num* = 10. In Statics2011, the AUC results are relatively steady. No large AUC gap is found, even for the small {5,10,1} and large {50,200,1} groups of the three parameters. It achieves the best results with only a few key memory slots and dimensions at {5,50,20}. Meanwhile, in Assist0910, the AUC gradually decreases with the increasing *s.dim* on each group of fixed *m.size* until *m.size* reaches 100 and obtains the best AUC at {50,10,1}. The different *s.dim* and *ri.num* settings do not lead to huge changes in the AUC results when the *m.size* is equal to 100. In Bridge2Algebra0607, the best AUC results in the group of the same *m.size* generally show an increasing trend with the increasing *m.size* until *m.size* reaches 100. The best result for this dataset is obtained at {50,200,10}. The AUC results gradually decrease with *m.size* = 100. In the Assist1213 dataset, the AUC increases with *s.dim* in each group of the same *m.size*. Moreover, for different *m.size*, the AUC shows an increasing trend until the *m.size* is equal to 100 and achieves the best result at {50,100,5}. The AUC decreases and does not change much around 73.10 when *m.size* reaches 100.

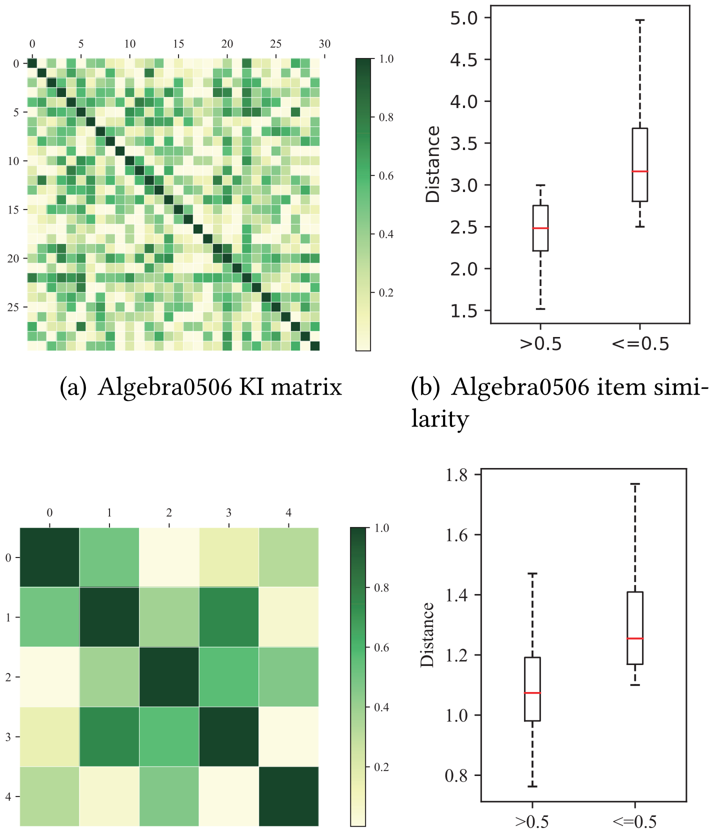
Fixed on the *m.size* and *s.dim* sets on each dataset, we further evaluate the sensitivity of different *ri.num* values (Figure 6.5). The result shows the impact of different numbers of the latest related items on the model performance evaluated by AUC, ACC, and Loss. In Assist0910, both AUC and ACC gradually decrease with the *ri.num* value increasing in the range of 1, 5, 10, 20, and 50, while the Loss first increases, decreases at 10, and then increases again at 20. It achieves the best AUC, ACC, and Loss when *ri.num* = 1. Assist0910 has the lowest average number of attempted items per learner and the lowest overall correctness, which is probably the reason why it is difficult for the model to capture the relation between each pair of latent KCs in the short sequences. For the other four datasets, both AUC and ACC first increase, and then decrease at a specific point. Furthermore, only one maximum point is found on the line, while the Loss sees an opposite trend. The best results are obtained at 10, 20, 10, and 5 for Algebra0506, Statics2011, Bridge2Algebra0607, and Assist1213, respectively.

We set the parameter set with the best results on each dataset to evaluate our proposed model based on the abovementioned observation.

B. Knowledge Interaction Between the Latent KCs

As mentioned before, we incorporate the knowledge interaction between the KCs into our model to capture the long-term dependencies in the exercising sequences and model the influence of the previous exercise on the current one. The KIEDCDA could automatically discover the underlying interaction between each pair of latent KCs, leading to a better model performance.

We directly visualize the final learned matrix M^I to verify how well the proposed model performs on discovering this knowledge interaction matrix. We further check the similarity of the items involving each pair of KCs, based on this matrix. Specifically, for each pair of KCs in the knowledge interaction matrix, we divide the KC pairs into



(a) Algebra0506 KI matrix

(b) Algebra0506 item similarity

(c) Statics2011 KI matrix

(d) Statics2011 item similarity

Figure 6.6: Knowledge interaction analysis on two datasets: (a) visualizes the learned knowledge interaction matrix of the first 30 KCs in the Algebra0506 dataset; (b) plots the similarity of each pair of items in Algebra0506 involving the pair of KCs, whose interaction coefficient is under or above a threshold of 0.5; (c) visualizes the knowledge interaction matrix of the five KCs in Statics2011; and (d) similarity of items in Statics2011 involving the pair of KCs, whose interaction coefficient is under or above a threshold of 0.5.

two parts: one with the interaction coefficient over a specific threshold and another under the same threshold. For each KC pair in these two parts, we collect two groups of items involving the pair of KCs based on the item–KC correction vector w_t obtained using Eq. (3), and calculate the Euclidean distance of the embedding vector k_t obtained using Eq. (2) for each pair of items in these two groups.

We take Algebra0506 and Statics2011 as examples for verifying the ability of our model to discover the knowledge interaction of KCs from the data. Figure 6.6 shows the results. Figures 6.6(a) and 6.6(c) indicate the knowledge interaction matrix of Algebra0506 and Statics2011, respectively. For a better illustration, we only show the first 30 KCs in the matrix among the 100 latent KCs in Algebra0506. We plot the item similarity involving a KC pair whose coefficient is over and under the threshold of 0.5 for both Algebra0506 and Statics2011, based on the learned matrix (Figures 6.6(b) and 6.6(d)). In both datasets, the items involving KCs with a coefficient over 0.5 in the matrix have a smaller distance than those with a coefficient under 0.5, because they are closer in the knowledge space. This result verifies that our model can effectively discover the knowledge interaction of KCs during the modeling process.

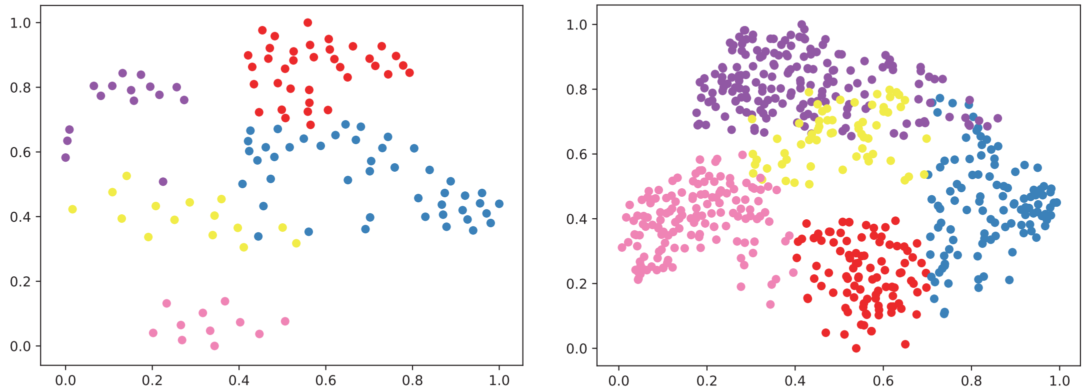
C. KC Discovery for Each Item

Section 4.3 showed that our model calculates the item–KC correction for each item to automatically discover the involved KCs therein. We directly visualize the learned embedding vectors of the items to demonstrate the model's ability to evaluate how well it discover the latent KCs for items. For a better illustration, we randomly select five KCs and all their corresponding items in a dataset and scatter the embedding vectors of these items in the learned embedding matrix A. Furthermore, we adopt t-SNE [159] in Python to visualize the high-dimensional data by projecting the high-dimensional embedding vectors of the items to two-dimensional points.

We take Algebra0506 and Statics2011 as examples to verify the ability of our model to discover KCs for items. We assign an item to a KC whose correlation value is the largest in the slot of the embedding vector for the item. Figure 6.7 depicts the scattering results for the two datasets. The items assigned to the same KCs are labeled with the same color. Consequently, the items with the same KCs are easier to be grouped because of their similarity in the knowledge space. The figure also reveals a compelling result when assigning items to the KCs, indicating the ability of our model to discover the KCs involved in the items.

6.5.3 Ablation Study

We conduct some ablation tests to examine the effectiveness of each model component. As mentioned before, our model can exploit the long-term dependencies of items,



(a) Algebra0506

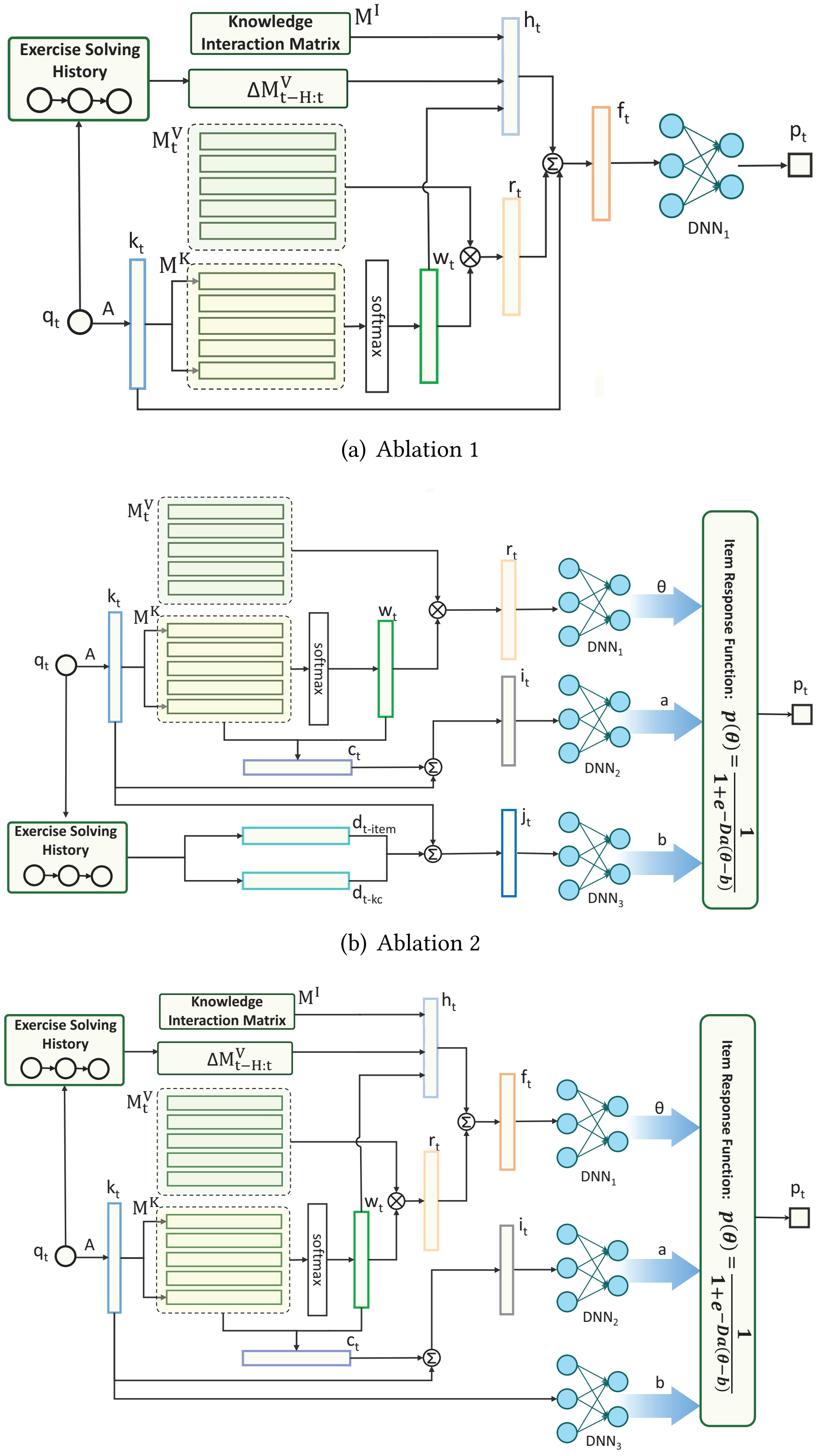
(b) Statics2011

Figure 6.7: Scattering results of items with five types of KCs in the Algebra0506 and Statics2011 datasets. The items involving different KCs are distinguished in different colors.

learner-oriented cognitive factor, and item parameters in the exercising sequences to model the learner performance, which leads to the best performance of the proposed model, by incorporating the knowledge interaction process and the cognitive item difficulty and extending the network using the IRT model. We propose herein three ablation models based on the KIEDCDA model to verify the contribution of each component (Figure 6.8).

- *Ablation 1*: Compared to the proposed model, this ablation model was incorporated the knowledge interaction process, but not extended with the IRT model (Figure 6.8(a)). It is used to test the influence of the IRT component.
- *Ablation 2*: This ablation model does not have the knowledge interaction component, and thus, cannot capture the long-term dependencies in the exercising sequences (Figure 6.8(b)). It is used for a performance comparison with the full model, considering the knowledge interaction.
- *Ablation 3*: This model uses the item embedding vector to infer the item difficulty and considers the item difficulty to be static for each item. Meanwhile, in the proposed KIEDCDA model, the learner-oriented cognitive item difficulty is used to adaptively calculate the item difficulty. Figure 6.8(c) illustrates this model.

We compare three ablation models with the proposed full model on all datasets.



(a) Ablation 1

(b) Ablation 2

(c) Ablation 3

Figure 6.8: Framework of the three ablation models: (a) ablation model without IRT extension; (b) ablation model without the knowledge interaction process; and (c) ablation model without the adaptive cognitive item difficulty.

Figure 6.9 illustrates the comparison results of AUC, ACC, and Loss. We discover that the KIEDCDA model generally outperforms all three ablation models on the five datasets based on the three metrics, verifying that considering the abovementioned three aspects leads to the best performance of the proposed model. Moreover, the results of the KIEDCDA model obtained for all datasets show lower variances, demonstrating its stability in modeling the learner performance. In the Algebra0506 dataset, the Ablation 2 model performs much worse than the other two ablation models, indicating the importance of considering the knowledge interaction in this dataset. In Statics2011, all three ablations have huge performance gaps compared to the KIEDCDA model. Moreover, Ablations 1 and 3 perform much worse than Ablation 2. This result implies that all three aspects considered in the KIEDCDA model are important for the final performance on this dataset, and that the item parameters and cognitive difficulty play more crucial rules. In Assist0910, the Ablation 3 model exhibits the worst performance among all models. Huge gaps exist between it and the other models, indicating that the learner-oriented cognitive item's difficulty matters much in this dataset. Moreover, the Ablation 2 model obtains results similar to those for the KIEDCDA model, which is consistent with the parameter analysis in Section 6.2.1, indicating that considering more previous items does not significantly improve the performance of this dataset. In Bridge2Algebra0607, Ablation 1 performs the worst, whereas Ablations 2 and 3 do not show very big differences, although they are still worse than the KIEDCDA model. These results indicate the huge influence of extending the model with IRT on this dataset when considering the item parameters. In Assist1213, the Ablation 2 and 3 models are much worse than the other models. Meanwhile, Ablation 1 shows results similar to those of the KIEDCDA model, indicating that the knowledge interaction and cognitive item difficulty play more crucial roles than IRT extension in this dataset.

In summary, all three considered aspects in the KIEDCDA model contribute to the best performance of the model on the five datasets and have different effects on different datasets.

6.5.4 Knowledge and Domain Modeling

As discussed in Section 1, the learners' knowledge proficiencies evolve over time with the exercising process. Moreover, our proposed KIEDCDA model can estimate the

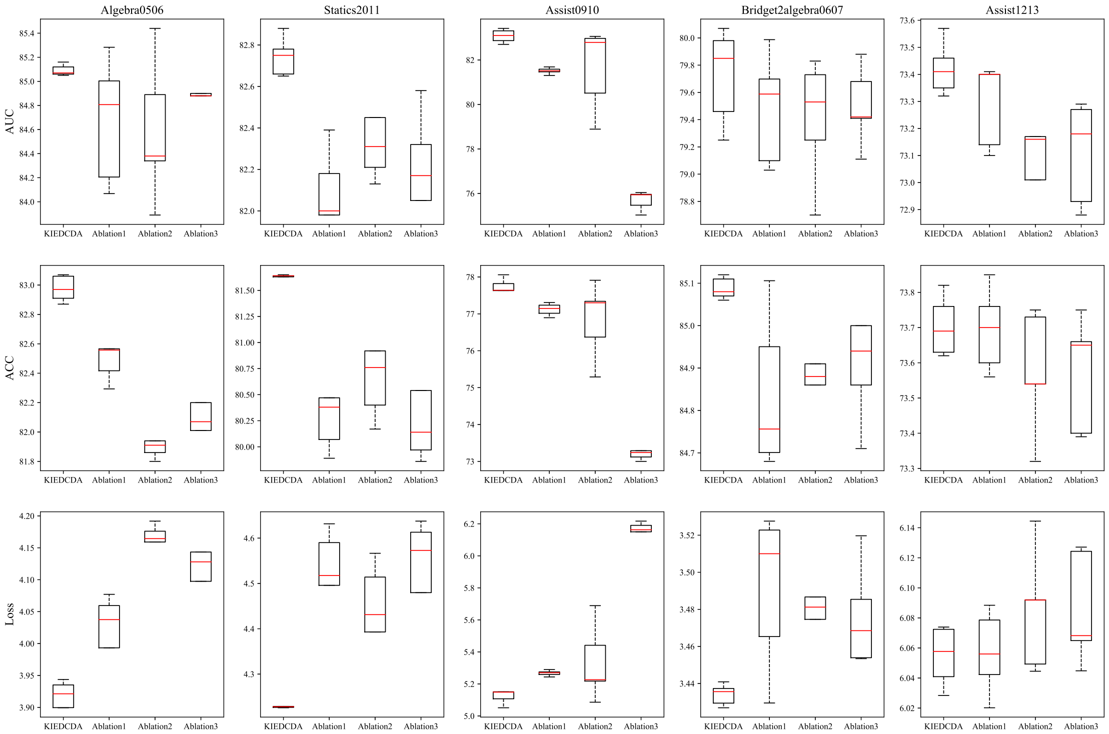


Figure 6.9: Comparison results of the three ablation models and the proposed model on five datasets based on three metrics.

multi-granularity knowledge proficiencies of learners and the item characteristics for modeling their performance, making it possible to delve deep into the causality between the learner performance and their proficiency, as well as item attributes.

We randomly select a learner from the Assist0910 dataset to evaluate the performance of our model on the modeling knowledge and item attributes in the domain. We then evaluate the evolution of his knowledge proficiency on 50 items. For a better illustration, we select items in his exercising sequence such that only five latent KCs are incorporated based on the item–KC correction. We test our model on these 50 items and five KCs (i.e., at each time step of the 50-item exercising process, a knowledge state comprised the knowledge proficiency of five KCs). After inputting this sequence into our model, it will infer the knowledge state after attempting each item and store it into the value memory M_t^V. Moreover, it will output the general ability of learners after each attempt and the item difficulty and discrimination by extending the IRT model. We divide the analysis into two parts: evolution of knowledge proficiency and skill

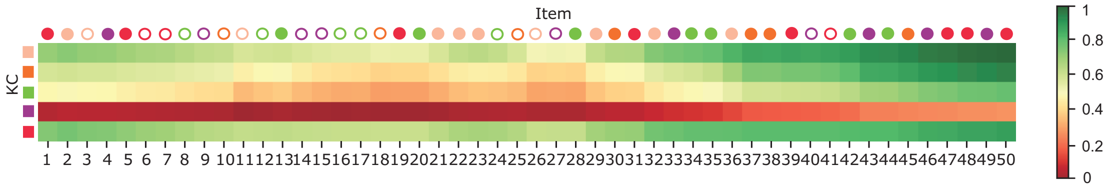


Figure 6.10: Evolution of KC proficiency during a learner's attempt on the 50 items containing five KCs. Different KCs are labeled with different colors. The items attempted correctly and incorrectly are represented with filled and hollow circles, respectively.

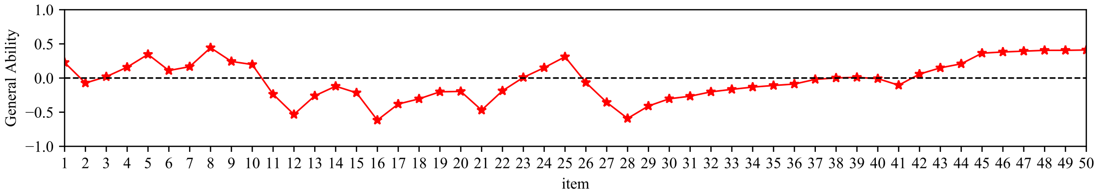


Figure 6.11: Evolution of the general ability during the learner's attempt on the items in Figure 6.10.

domain analysis.

A. Evolution of Multi-granularity Knowledge Proficiency

To show a learner's knowledge proficiency on each KC at each time step, we directly input the i_{th} vector in the value memory into the network DNN_1 to output the value θ_i of the proficiency on the i_{th} KC using the following formula:

$$\theta_i = sigmoid(DNN_1(M_i^V)) \tag{6.23}$$

Figure 6.10 depicts the results of the proficiency evolution of the five KCs when attempting 50 items. We obtain the general ability of the learner after attempting each item using the IRT in Eq. (20) (Figure 6.11), making it possible to present the learner knowledge in a multi-granularity manner.

Figure 6.10 shows that the evolution of the KC proficiency is very smooth when the 50 items are attempted. Some compelling results are also obtained. After initializing the knowledge state for each KC before attempting any item, our model discovers the KCs from each attempted item and updates the state for the discovered KCs at each time step. For example, when the learner answers the first item correctly, the state value of the fifth KC (marked red) underlying this item increases. When the learner obtains incorrect answers for the sixth and seventh items, the state value of the fifth KC gradually decreases. When when he attempts the 19th, 31st, and 39th items with the same fifth KC correctly, this value increases again until it shows that the learner has mastered this KC. Another example is the second KC with the orange color. When the learner incorrectly answers the 10th, 18th, and 25th items, the proficiency value of the second KC gradually decreases. After the learner answers the 30th, 37th, 38th, and 45th items correctly, the value begins to increase until a moderate level. Figure 6.11 shows that during the first 28 attempts, the learner attempts to correctly and incorrectly answer the items in a crosswise manner; hence, the general ability frequently fluctuates. After the learner attempts most of the remaining items correctly, the wave of the general ability gradually rises. In summary, the results of the KC proficiency evolution present the learner's knowledge level evolution on each KC at a different time step, while the general ability evolution illustrates the learner's general ability for all KCs during the exercising process. All results show a reasonable consistency with the learner's real responses in the exercising process, indicating the ability of our KIEDCDA model to estimate the multi-granularity knowledge state during the learning process.

B. Skill Domain Analysis

When the KIEDCDA model is incorporated with IRT, it cannot only output the learners' knowledge proficiency but also obtain the estimated item characteristics in the skill domain, making it interpretable for explaining the learners' performance in terms of their current knowledge and item characteristics.

We visualize herein the same sets of items in Figure 6.10 to present the obtained item difficulty and discrimination (Figures 6.12 and 6.13). Figure 6.12 compares the results of the three methods for calculating the item difficulty. The difficulty by item is achieved using Eq. (11) based on the same items as those used in the previous attempts. The difficulty by the KC is calculated using Eq. (12) based on the same underlying KCs in the previous attempts. Meanwhile, the cognitive item difficulty is proposed

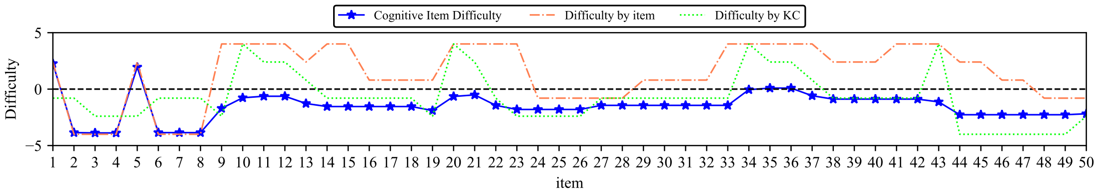


Figure 6.12: Comparisons of the three types of difficulties for the items shown in Figure 6.10.

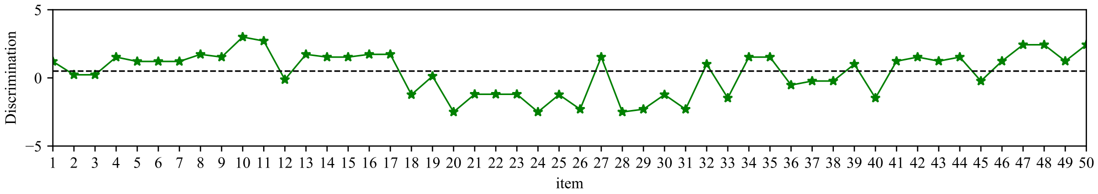


Figure 6.13: Obtained item discrimination for the attempted items in Figure 6.10.

herein by considering the previous attempted items and KCs and the current item in an adaptive manner, and is calculated using Eq. (13). For a comparison, we also normalize the results of the difficulties by both item and KC into [-4,4] as the same scale with the cognitive item difficulty. Figure 6.12 depicts that the difficulty by item generally has a much larger value because learners do not frequently interact with the same items. Moreover, the initial parameter setting for a new item is the highest level of difficulty; hence, this line is presented at a higher position. In contrast, the difficulty according to the KC frequently changes at a much lower position because learners may frequently come across the same KC by attempting items with the same KCs. Comparatively, the cognitive item difficulty shows much stable changes during the exercising process and maintain the overall trend with the other two lines for each item.

Figure 6.13 shows that the item discrimination frequently changes because different items contain different KC proportions; hence, different discriminatory attributes are presented. We also discover that many items have very low discrimination, which is not good for adaptive tutoring because these items cannot classify high- and low-ability learners.

In summary, our KIEDCDA model can output the item characteristics in the domain, thereby providing clues for exploring the learner performances. Moreover, based on the domain analysis, we can further provide suggestions to system builders with regard to improvement in their tutoring systems, by selecting items for learners for better adaptive learning. Items that are estimated too difficult or too easy and are not in conformity with the learners' knowledge level can be skipped or delayed, and those with no good discriminatory attribute can be deleted. Therefore, the learners' learning efficiency can be effectively improved and any decrement in their engagement can be avoided.

6.6 Summary

This chapter looks into **Issue 3**: *how to track and explain learners' evolving knowledge states simultaneously?* It presented a new CDA model, called KIEDCDA, for diagnosing the knowledge proficiency of learners by modeling their learning performances. KIEDCDA unified the key-value memory network and IRT model into the model framework to dynamically trace the evolution of learners' knowledge proficiencies in a long period and enable interpretability to explain learner performances. This chapter proposed the knowledge interaction concept among the knowledge concepts underlying the items and incorporated it into the proposed model to capture the long-term dependencies of the items in the exercising sequences. Moreover, the cognitive item difficulty was adaptively modeled to more precisely obtain the learner-oriented item difficulty. Using these factors, the KIEDCDA model can output not only the multi-granularity knowledge proficiency of each learner but also the item characteristics, thereby providing clues for explaining the learner performances in terms of their knowledge proficiencies and the characteristics of the attempted items. Finally, extensive experiments were conducted on five large-scale, real-world datasets and evaluated the proposed model from six perspectives. The results demonstrated the effectiveness and interpretability of the model.

7

Conclusion

In this thesis, we have explored the task of dynamic learner knowledge assessment to obtain the individual learner's evolving knowledge states, which indicate the mastery of the particular knowledge in a domain, based on the massive long-term learning logs in the ITSs. In the previous chapters, we have proposed three different approaches to solve the three existing issues from different perspectives. In this chapter, we summarize our work and provide reply to the three issues in Section 7.1. In Section 7.3, we discuss the remaining issues for the task of DLKA and present the future work.

7.1 Replies to The Three Issues of DLKA

In this work, a general framework that is used as a general idea for solving the task of DLKA is presented. This framework assess learner knowledge by incorporating both learner and domain modeling. Compared with the existing work that generally performs learner knowledge assessment based on the exercising results, this framework assesses learner knowledge by considering multiple factors that not only related with the exercising results but also the exercising procedures. This framework is then instantiated into three approaches that address the three following issues from different perspectives: insufficient learning factor modeling, data sparseness and information loss, fine-grained assessment and interpretability. Here we give reply to these three issues and describe in detail how the three proposed approaches solve these issues.

Issue 1*: what factors influence the learning performance and how to quantify these factors and utilize them to model the dynamic evolution of learner knowledge?*

The learning performance of learners is generally related with many factors because of the extremely complicated human knowledge construction procedure. During the long-term exercising procedure, learners update their knowledge incessantly by interacting with the exercise, hence these factors that influence the learning performance should not only include the learner factors but also the domain factors.

In Chapter 4, we explored this issue and investigated the learner factors (learning and forgetting) and domain factor (item difficulty) by making use of rich information during learners' learning interactions to achieve more precise prediction of learner knowledge. Specifically, we proposed a novel model named KTM-DLF that traces the evolution of learners' knowledge acquisition over time by explicitly modeling learners' learning and forgetting behaviors as well as the item difficulty. Based on two classical theories (the learning curve theory and the Ebbinghaus forgetting curve theory), we proposed methods for modeling learners' learning and forgetting behaviors by taking account of their memory decay and the benefits of attempts on exercises. We also specified the concept of cognitive item difficulty and proposed a method to model this user-oriented difficulty adaptively in terms of the cognitive challenge it presents to different individuals. The KTM-DLF model is then proposed to incorporate learners' abilities, the cognitive item difficulty, and the two dynamic procedures (learning and forgetting) together. To further increase the model's accuracy, the factorization machine framework was utilized to embed features in high dimensions and model pairwise interactions among these features.

Compared with existing studies that consider only a fragment of the information related with learning or forgetting and almost all work that either neglects the problem difficulty or assumes that it is constant, the KTM-DLF model takes more and precise information into the modeling procedure. Extensive experiments confirmed the effectiveness of our proposed model. Ablation studies were also conducted to test each considered factor, and the results showed that all these factors contributed to the improvement of model performance on the learner knowledge assessment.

Issue 2*: How to alleviate the data sparseness and the information loss in conducting learner knowledge assessment?*

Learner knowledge assessment methods have achieved good performance at this task. However, the adequacy of model performance is still challenged by the sparseness of the learners' exercise data as students are not required to answer all the questions in an ITS, meaning that some students may not answer some questions. Moreover, each question is correlated with one or several skills needed to solve the question. Accordingly, the response data is quite sparse. Existing studies implement their models at the skill-level rather than the question-level, hence the distinctive information related to the questions themselves and their relations are neglected, which has caused the potential information loss. Due to the data sparseness and the information loss, the models can imprecisely infer the learners' knowledge states and might fail to capture the long-term dependencies in the exercising sequences.

To solve this issue, Chapter 5 explored to incorporate the knowledge structure (KS) into the learner knowledge assessment procedure to potentially resolve both the sparseness and information loss, an avenue not yet been fully explored because obtaining the complete KS of a domain is challenging and labor–intensive. We proposed a novel KS-enhanced graph representation learning model for KT with an attention mechanism (KSGKT). Specifically, we first explored eight methods that automatically infer the domain KS from learner response data and integrate it into the KT procedure. The integration of KS into KT procedure offers two advantages over the existing models: first, it inputs extra information into the question representation by referencing the KS as a question requiring a specific skill is also related to the prerequisite skills, thus alleviating the data sparsity problem in the question representation; and second, it models the impact of previous experiences on future exercise during the knowledge evolution by referencing the KS because incorporating the KS into the KT procedure can capture the long-term dependencies in the exercising sequences. Moreover, it considered more factors in the learning domain that can be leveraged to monitor the evolution of learner knowledge, this further improves the precision of inferring the dynamic knowledge states of learners.

To alleviate the information loss, we applied a graph representation learning method (Metapath2Vec) to obtain question- and skill-embedding from the KS enhanced question–skill relation graph by leveraging the high ability of graph neural networks to extract graph representation by aggregating the information from neighbors. The learned embeddings from the graph incorporate not only the explicit multi-hop question–skill relations but also the implicit multi-hop question–question and skill–skill relations in the graph. To overcome the limitations of skill–level KT models, which neglect the distinctive information related to the questions, we also proposed a convolutional representation method that incorporates additional information and considers their interactions, thus generating dense and comprehensive representations of the input questions and potentially further improving the model performance. These representations are input to the proposed KT model, and the long-term dependencies are handled by the attention mechanism. The model finally predicts the learner's performance on new problems.

Extensive experiments conducted from six perspectives on three real-world datasets demonstrated the superiority and interpretability of our model for learner-performance modeling. We also tested the graph embedding learning to the model performance and showed the visualization of these embeddings, the results showed the effectivenss of this embedding learning procedure, and the embedding can definitely overcome the data sparseness and information loss. The visualization of the inferred KS also showed meaningful and interpretable results. Moreover, the proposed three attention-calculation methods also improved the model performance.

To sum up, the proposed method is a good trial to alleviate the data sparseness and the information loss in conducting learner knowledge assessment.

Issue 3*: how to track and explain learners' fine-grained and evolving knowledge states simultaneously?*

The existing models are either designed for static scenarios or find it difficult to explain the causality between learner performance and knowledge proficiency, as well as the item characteristics. CDA models have good interpretability because of the rich background educational psychology theories, but they are generally designed for the static assessment, and are difficult to meet the requirements of large-scale assessment. KT models dynamically track the evolution of learner knowledge, but most of the high-performance KT models are based on the deep neural networks and find it difficult to explain the results. Hence how to track and explain learners' evolving knowledge states simultaneously remains to be an important issue.

To solve this issue, in Chapter 6 we proposed a dynamic CDA model called KIEDCDA that incorporates not only the ability to trace the evolution of learners' knowledge proficiencies over time for large-scale assessments, but also the interpretability to explain learner performance in terms of their current knowledge proficiency and item characteristics. Specifically, We first proposed a dynamic CDA framework by unifying the strength of the memory capacity of the key-value memory network to enhance the representation of the knowledge state during learner performance modeling and the interpretability of the Item Response Theory to explain the learner performance in terms of knowledge proficiency and item characteristics (i.e., item difficulty and discrimination). In this framework, we traced each learner's knowledge proficiency on each knowledge concept over time and stored them into an auxiliary memory using the key-value memory network. We further inferred their general proficiencies and the IRT-based item characteristics using another neural network. Moreover, we proposed the knowledge interaction concept among KCs and incorporated it into the CDA procedure to further exploit the long-term dependencies in the exercising sequences, thereby devising the KIEDCDA model. Based on these factors, our KIEDCDA model could not only output the learners' knowledge proficiency in a multi-granularity manner but also output the item characteristics, making it possible to interpret the learner performances in terms of their current knowledge states and item characteristics.

Extensive experiments conducted from six perspectives on five real-world datasets demonstrated the superiority and interpretability of our model for learner performance modeling, suggesting that it is worthy of a good trial to track and explain learners' fine-grained and evolving knowledge states simultaneously.

7.2 Summary of Research Contributions to The ITS

In the above section, we reply to the three important issues solved in this thesis and show our contributions to the topic of DLKA. In this section we will summarize the research contributions of this thesis to the field of ITS from the whole picture.

- **"Stupid" Systems, Intelligent Tutoring—*Who They Teach?***

  As an alternative to the "one-size-fits-all" traditional web-based learning platforms, ITSs are designed to mimic individualized human tutoring in a computer-based environment [20], and are expected to offer delicate instructions during learners' learning process. ITSs used at scale today are developing rapidly to fill this expectation, but they are still computer programs (sometimes even stupid and simple programs), hence the ability to provide intelligent and adaptive tutoring services is essential for the widely applications in the daily life.

  This thesis explored the task of dynamic learner knowledge assessment to obtain the individual learner's evolving knowledge states, which is the pillar of learner characteristics in ITS. The distribution of a learner's knowledge states provides a distinctive latent profile of the learner for the ITSs, and lets the ITSs know who they are teaching, hence increasing the adaptability and individualization of the further services.

  Moreover, this thesis solves the task of dynamic learner knowledge assessment by modeling the individual knowledge acquisition procedure (learning and forgetting process) and obtaining the individual knowledge evolution, providing more accurate and individual information for the ITSs. This process of learner modeling can be directly integrated into the various existing online learning platforms (e.g., MOOCs, Massive Open Online Courses) to enhance their intelligent ability.

- **Learning Content Management in ITSs—*What They Teach?***

  Learning content is another important factor for the success of ITSs. Learning content is generally manually organized into curriculums with some structures (hierarchies, networks, frames, etc.) that link the knowledge together according to pedagogical sequences. The modeling of learning content provides the ITSs with the knowledge of what they are teaching. Learning content is generally labeled and structured by experts. Because of the labor-intensive work, the current ITSs usually store a limited amount of learning content in their item bank.

  To improve the scalability of the ITS to incorporate the overwhelmingly large numbers of learning items emerging on the Internet, it is essential to automatically label and structure the learning content. Obtaining the characteristics of learning content is fundamental for the management of learning content. This thesis explored the learning domain modeling and proposed to automatically obtain various aspects for the learning contents. In Chapter 4, it specified the concept of cognitive item difficulty and propose a method to model the cognitive item difficulty adaptively based on learners' learning histories. Compared with existing studies that either do not consider problem difficulty or assume it to be constant (i.e. item-oriented difficulty), the cognitive item difficulty considering the cognitive difficulty of items adaptively for different learners (i.e. user-oriented difficulty) will make the modeling of DLKA more accuracy for specific individuals. In Chapter 5, it explored eight methods to automatically discover the domain knowledge structure from the learner response data and test them in the knowledge tracing procedure. And the experimental results also confirmed that the knowledge structure discovery methods can infer reasonable prerequisite skill pairs from the ordering of learners' mastery of skills. Chapter 6 proposed a framework to output the estimated item characteristics (item difficulty and discrimination) in the learning domain. Moreover, the learned item/skill embeddings in Chapter 5 and Chapter 6 can be used to measure the similarity of learning content.

The obtained characteristics of learning content are essential for the ITSs to manage learning content and help the ITSs understand what they are teaching.

- **Interpretability of Decision-making in ITSs—*Potential for Explainable Feedback***

Interpretability is of great importance for the tutoring in ITSs. Learners generally not only expect the ITSs to provide the adaptive tutoring but also want to know why the specific kind of tutoring (e.g., recommend an item) is provided. The general deep learning based models in ITSs are deemed as black boxes, and it is difficult to explain the decision-making based on such models.

In this thesis, we explored to track and explain learners' evolving knowledge states simultaneously. Our final aim is to provide the learners with the tailored learning content that can remedy their weak knowledge concepts and improve their learning efficiency based on the result of dynamic learner knowledge assessment. In Chapter 5, we proposed to generate a diagnostic report of learners' mastery on each skill over time based on the hidden vectors in the deep learning framework. This provides some degree of interpretability for the distribution of learner knowledge, and also further clues to explain why some items are further recommended to the learners. Chapter 6 proposed a model that can not only output the learners' knowledge proficiencies in a multi-granularity manner but also output the item characteristics, thereby providing clues for explaining the learner performance prediction in terms of their knowledge proficiencies and the characteristics of the attempted items.

To summarize, this thesis investigated the potential of educational data mining driven decision-making in ITSs for adaptive online tutoring, and provided preliminary interpretability for the results. This will be helpful for the further tutoring services and provides ideas for further explainable feedback.

7.3 Remaining Issues and Future Work

Although the proposed methods in the previous chapters and existing work have obtained good results for the task of dynamic learner knowledge assessment, there are still some gaps when applying them for the practical applications. Here we list some of them, which we want to fill in the future work.

- **Q-matrix Learning**

  In this thesis, we assume that the KC set and the Q-matrix, which maps the questions to the KCs in the domain, are already given by the experts. Actually this is also a basic assumption of previous studies. The ITSs generally keep a certain amount of learning materials in their databases with fine-labeled structure to practice certain set of KCs in the domain. To obtain the set of elicited KCs and the labelled Q-matrix, large amount of the manual work from expert should be conducted, which is highly time-consuming and labor-intensive, particularly for a complicated subject with a large quantity of knowledge [18]. Nevertheless, there is severe issue of consistency as trade-offs should be made between granularity and coverage.

  For the estimation of model parameters and the identification of the underlying knowledge states of learners, the appropriately elicited KCs and defined Q-matrix are essential. Previous research showed that the learner knowledge assessment methods are susceptible to Q-matrix choices. A misspecified Q-matrix might lead to significant inadequate fitness and hence erroneous identification of learner knowledge states [76]. On the other hand, in this information era, there are increasing numbers of open materials in the Internet that can be reused by these ITSs, automatically structuring,

Table 7.1: Various modalities of data that can be used for learner knowledge assessment.

Modality	Usage for learner knowledge assessment
Exercise text	KC elicitation, Q-matrix designing, attention mechanism, domain modeling
Video watching, clickstream sequences, readings, forum discussion	Explore learning patterns, predict future performance
IoT data	Learning behavior, affective and cognitive factors, learning style and preference, motivation, disengagement
Demographic data	Learner modeling

assessing and labeling these open materials will greatly enhance the intelligence and scalability of existing systems. Hence the automated methods for the KC elicitation and Q-matrix designing have become a prominent necessity in the intelligent tutoring field.

Some researchers have explored to learn the Q-matrix and the KC set automatically from the learning data [76, 66]. However, these methods obtain the Q-matrix with unknown KCs, thus making them difficult to interpret as expert-made and the inferred Q-matrices do not often coincide. To improve the interpretability, automated Q-matrix learning methods by exploiting both the student performance data and the side information (e.g., text) of learning items could be a promising direction, which may be worthy of trial in the future.

- **Multi-modal Learner Knowledge Assessment**

This thesis conducted the learner knowledge assessment mainly based on the long-term gradable exercising logs collected in the ITSs. However, this is only one typical modality of learning data that can be used for this task, and the other modalities of data (as shown in Table 7.1) are largely ignored, with which the model performance and the intelligence of ITSs could potentially be further improved.

In this thesis, the exercise text is not been used for the proposed models even though they all take the exercising logs as input. Actually, there is rich information contained in the exercise text, e.g., KCs, exercise hierarchy, difficulty and discrimination. Leveraging natural language processing methods, the KCs contained in the exercises can be elicited, as in [90]. The exercise text can also facilitate the designing of Q-matrix, and joining the data-driven and the text analysis methods will be a feasible direction for obtaining an interpretable Q-matrix. Moreover, the exercise text can be also used to distinctively represent the certain exercise even with the same KCs, which can be used for the item representation and attention calculation [1]. Despite these benefits, the exercise text has not been used in this thesis, as most of the open datasets do not provide the text information of the exercises in their systems. With the popularity of multimodal learning analytics, some datasets with text information are becoming open-access. Hence in the future, we plan to combine both learner exercising logs and the exercise text for more precisely assessing learner knowledge.

Moreover, the practical learning procedure is conducted not only on the exercises, but also some other various learning materials, such as the videos and readings, especially on the popular MOOCs platform. Hence some other modalities of data is widely existed. These video watching and reading logs, clickstream sequences and forum discussion data can be additional fuel to improve the learner knowledge assessment models [45, 44, 46, 47]. In addition, the demographic data of learners can be also used for learner modeling.

Learning analytics researchers have to date depended on ITS data to examine learners' knowledge states. While these data sources can still provide a rich ground for learner knowledge assessment, a new wave of technological innovations is taking place with the Internet of Things (IoT). Wearables, eye-trackers and other camera systems provide new physiological sources of information, hence multimodal datasets can be collected from physical activities and physiological responses to learning situations that may be utilized to investigate and assist learning [167, 168]. The development of integrating wearable computing and learning analytics using both physiological and learning log data collected as students interact with learning content is still marginal. This level of multimodal data collection promises to provide valuable insight into cognitive and affective states [169], and also the knowledge states of individuals, especially when combined with traditional learning analytics data sources, and will definitely provide both new opportunities and challenges to enhance our understanding of the learning process and employ these insights to intelligent tutoring better.

- **Learner Knowledge Assessment Beyond Binary Correctness**

  In this thesis, we use the binary correctness (1/0) to show whether the learners attempt exercises correctly or incorrectly. This is a natural way to indicate the learning performance on objective exercises. Following the existing work, the performance on subjective exercises (e.g., the programming exercises) are ignored or simplified as binary correctness when building the models, hence the information from subjective problems is largely underexplored. Actually subjective exercises are widely used in the practical learning procedure. Compared with the objective exercises, the response to a subjective exercises can be continuous values, ranging from totally correct, totally wrong, and partially correct [33]. Liu et al. [33] argued that subjective exercises measure the learners much better as learners may be unable to guess the exercises correctly to a large degree.

Some researches explored to use such sources of richer information, for example [170] made the binary correctness continuous using partial credit. However, such kind of methods has still not been used in the current mainstream of learner knowledge assessment. A most recent work [171, 172] compared binary correctness with non-binary measures and showed that the binary correctness as the only input to the models was not always warranted and limited the progress of ITSs. Hence it is necessary to develop models that can assess learner knowledge based on both discrete and continuous performance data.

- **Learner Knowledge Assessment with Small Learning Data**

  As described in Chapter 3, the models in this thesis used several datasets that include big volume of learning logs collected from the real-world ITSs. These big educational data enables the good performance and stability of the proposed models, especially for the deep neural network models. However, the model performance may be degraded when applied to the small learning data, especially the classroom-level data, as Inadequate sample size can result in poor parameter estimates with poor predictive power.

  The traditional cognitive diagnostic models, such as IRT and DINA, are designed for this small-data context, they are widely used for the classroom-level test with a certain number of exercises associated with small number of KCs (usually less than ten), but they are designed for the static assessment and the temporal information is not considered. The powerful knowledge tracing models can dynamically monitor the learner knowledge but most of them are built on the massive data. Hence the current learner knowledge assessment model are rarely seen for the practical classroom usage, which greatly limit the further application of these models. It is necessary to propose approaches to conduct learner knowledge assessment based on small data.

We assume that two potential solutions can be explored in the future to cope with the small data. One is to infer and tune the model parameters on the big data and use the trained model to conduct prediction on the small data. The other is based on the Variational Autoencoders (VAE) [173], which is a famous generative model. The VAE model can learn the probability distribution parameters of the input sample in order to generate new data that are similar to those in the small dataset. We will validate these ideas in our future work.

- **Further Tutoring Based on Learner Knowledge Assessment**

  In this thesis, we modeled learner performance to infer their latent knowledge states by integrating learner and domain modeling. The learner factor (knowledge states) and domain factors (item difficulty and discrimination, knowledge graph) can be obtained from the output of these models. However, how to further utilize these factors to provide adaptive tutoring services has yet been conducted.

  Actually providing the intelligent tutoring services to individual learners is the final objectives for developing all kinds of learning analytics techniques. This thesis is part of the work towards this direction. Some researchers have explored the learning material recommendation based on the inferred knowledge states [53, 174]. Different from the traditional collaborative filtering-based recommendation, the knowledge states based methods provide more personalization, which is worthy of trial in the future. Moreover, we also plan to conduct case studies to test the effectiveness of integrating learner knowledge assessment and online learning systems on student learning performances and perceptions.

This thesis presented our trials on dynamic learner knowledge assessment, we hope it will stimulate new ideas in the field of intelligent education that will overcome all educational barriers in the COVID-19 era and serve every individual learner adaptively.

Bibliography

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

[2] Zhenya Huang, Qi Liu, Yuying Chen, Le Wu, Keli Xiao, Enhong Chen, Haiping Ma, and Guoping Hu. Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–33, 2020.

[3] Albert T Corbett, Kenneth Koedinger, and William S Hadley. Cognitive tutors: From the research classroom to all classrooms. In *Technology enhanced learning*, pages 215–240. Routledge, 2001.

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

[5] Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 341–344, 2020.

[6] Jijuan Cao, Ting Yang, Ivan Ka-Wai Lai, and Jun Wu. Student acceptance of intelligent tutoring systems during covid-19: The effect of political influence. *The International Journal of Electrical Engineering & Education*, page 00207209211003270, 2021.

[7] Florence Martin, Ting Sun, and Carl D Westine. A systematic review of research on online teaching and learning from 2009 to 2018. *Computers & education*, 159:104009, 2020.

[8] Elham Mousavinasab, Nahid Zarifsanaiey, Sharareh R. Niakan Kalhori, Mahnaz Rakhshan, Leila Keikha, and Marjan Ghazi Saeedi. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. *Interactive Learning Environments*, 29(1):142–163, 2021.

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

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

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

[12] Wenbin Gan, Yuan Sun, Xian Peng, and Yi Sun. Modeling learner’ s dynamic knowledge construction procedure and cognitive item difficulty for knowledge tracing. *Applied Intelligence*, 50(11):3894–3912, 2020.

[13] Wenbin Gan, Yuan Sun, and Yi Sun. Knowledge interaction enhanced knowledge tracing for learner performance prediction. In *2020 7th International Conference on Behavioural and Social Computing (BESC)*, pages 1–6. IEEE, 2020.

[14] Aziz Hasanov, Teemu H Laine, and Tae-Sun Chung. A survey of adaptive context-aware learning environments. *Journal of Ambient Intelligence and Smart Environments*, 11(5):403–428, 2019.

[15] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao. Online learning: A comprehensive survey. *arXiv preprint arXiv:1802.02871*, 2018.

[16] Michel C Desmarais and Ryan S Baker. A review of recent advances in learner and skill modeling in intelligent learning environments. *User Modeling and User-Adapted Interaction*, 22:9–38, 2012.

[17] José Hanham, Chwee Beng Lee, and Timothy Teo. The influence of technology acceptance, academic self-efficacy, and gender on academic achievement through online tutoring. *Computers & Education*, page 104252, 2021.

[18] Hyacinth S Nwana. Intelligent tutoring systems: an overview. *Artificial Intelligence Review*, 4(4):251–277, 1990.

[19] John A Self. Student models in computer-aided instruction. *International Journal of Man-machine studies*, 6(2):261–276, 1974.

[20] Ekaterina Kochmar, Dung Do Vu, Robert Belfer, Varun Gupta, Iulian Vlad Serban, and Joelle Pineau. Automated data-driven generation of personalized pedagogical interventions in intelligent tutoring systems. *International Journal of Artificial Intelligence in Education*, pages 1–27, 2021.

[21] Ronnie Cheung, Calvin Wan, and Calvin Cheng. An ontology-based framework for personalized adaptive learning. In *International Conference on Web-Based Learning*, pages 52–61. Springer, 2010.

[22] Roger Nkambou, Riichiro Mizoguchi, and Jacqueline Bourdeau. *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media, 2010.

[23] Radek Pelánek. Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3-5):313–350, 2017.

[24] Shan Li, Susanne P Lajoie, Juan Zheng, Hongbin Wu, and Huaqin Cheng. Automated detection of cognitive engagement to inform the art of staying engaged in problem-solving. *Computers & Education*, 163:104114, 2021.

[25] Juan Feldman, Ariel Monteserin, and Analía Amandi. Automatic detection of learning styles: state of the art. *Artificial Intelligence Review*, 44(2):157–186, 2015.

[26] Kiavash Bahreini, Rob Nadolski, and Wim Westera. Towards multimodal emotion recognition in e-learning environments. *Interactive Learning Environments*, 24(3):590–605, 2016.

[27] Jinseok Lee and Dit-Yan Yeung. Knowledge query network for knowledge tracing: How knowledge interacts with skills. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 491–500, 2019.

[28] Yuying Chen, Qi Liu, Zhenya Huang, Le Wu, Enhong Chen, Runze Wu, Yu Su, and Guoping Hu. Tracking knowledge proficiency of students with educational priors. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 989–998. ACM, 2017.

[29] Benoît Choffin, Fabrice Popineau, Yolaine Bourda, and Jill-Jênn Vie. DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills. In *Proceedings of the 12th International Conference on Educational Data Mining (EDM)*, pages 29–38, 2019.

[30] Jill-Jênn Vie and Hisashi Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.

[31] Ghodai Abdelrahman and Qing Wang. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2019.

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

[33] Qi Liu, Runze Wu, Enhong Chen, Guandong Xu, Yu Su, Zhigang Chen, and Guoping Hu. Fuzzy cognitive diagnosis for modelling examinee performance. *ACM Transactions on Intelligent Systems and Technology*, 9(4):1–26, 2018.

[34] Wenbin Gan, Yuan Sun, Shiwei Ye, Ye Fan, and Yi Sun. AI-Tutor: Generating tailored remedial questions and answers based on cognitive diagnostic assessment. In *2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC)*, pages 1–6. IEEE, 2019.

[35] Panagiotis Adamopoulos. What makes a great mooc? an interdisciplinary analysis of student retention in online courses. In *Proceedings of the 34th International Conference on Information Systems*, pages 1–21. Association for Information Systems, 2013.

[36] Zhuo Wang, Jile Zhu, Xiang Li, Zhiting Hu, and Ming Zhang. Structured knowledge tracing models for student assessment on coursera. In *Proceedings of the Third ACM Conference on Learning at Scale*, pages 209–212. ACM, 2016.

[37] Zachary A Pardos, Yoav Bergner, Daniel T Seaton, and David E Pritchard. Adapting bayesian knowledge tracing to a massive open online course in edx. In *Proceedings of the 6th Annual International Conference on Educational Data Mining*, pages 137–144, 2013.

[38] Lu Guo, Dong Wang, Fei Gu, Yazheng Li, Yezhu Wang, and Rongting Zhou. Evolution and trends in intelligent tutoring systems research: a multidisciplinary and scientometric view. *Asia Pacific Education Review*, pages 1–21, 2021.

[39] Danielle S Bassett and Marcelo G Mattar. A network neuroscience of human learning: potential to inform quantitative theories of brain and behavior. *Trends in Cognitive Sciences*, 21(4):250–264, 2017.

[40] Radek Pelánek. Managing items and knowledge components: domain modeling in practice. *Educational Technology Research and Development*, 68(1):529–550, 2020.

[41] Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. Gikt: A graph-based interaction model for knowledge tracing. *arXiv preprint arXiv:2009.05991*, 2020.

[42] Yunfei Liu, Yang Yang, Xianyu Chen, Jian Shen, Haifeng Zhang, and Yong Yu. Improving knowledge tracing via pre-training question embeddings. *arXiv preprint arXiv:2012.05031*, 2020.

[43] Ryan Shaun Baker, Albert T Corbett, Kenneth R Koedinger, and Angela Z Wagner. 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*, pages 383–390, 2004.

[44] Khushboo Thaker, Paulo Carvalho, and Kenneth Koedinger. Comprehension factor analysis: Modeling student's reading behaviour: Accounting for reading practice in predicting students' learning in moocs. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 111–115, 2019.

[45] Ahmed Ali Mubarak, Han Cao, and Salah AM Ahmed. Predictive learning analytics using deep learning model in moocs' courses videos. *Education and Information Technologies*, 26(1):371–392, 2021.

[46] Yu-Chen Chiu, Hwai-Jung Hsu, Jungpin Wu, and Don-Lin Yang. Predicting student performance in moocs using learning activity data. *J. Inf. Sci. Eng.*, 34(5):1223–1235, 2018.

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

[48] Tsung-Yen Yang, Christopher G Brinton, Carlee Joe-Wong, and Mung Chiang. Behavior-based grade prediction for moocs via time series neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 11(5):716–728, 2017.

[49] Heonseok Ha, Uiwon Hwang, Yongjun Hong, Jahee Jang, and Sungroh Yoon. Deep trustworthy knowledge tracing. *arXiv preprint arXiv:1805.10768*, 2018.

[50] Georg Rasch. *Studies in mathematical psychology: I. Probabilistic models for some intelligence and attainment tests.* Nielsen & Lydiche, 1960.

[51] Hanshuang Tong, Yun Zhou, and Zhen Wang. Hgkt: Introducing problem schema with hierarchical exercise graph for knowledge tracing. *arXiv preprint arXiv:2006.16915*, 2020.

[52] Daqian Shi, Ting Wang, Hao Xing, and Hao Xu. A learning path recommendation model based on a multidimensional knowledge graph framework for e-learning. *Knowledge-Based Systems*, 195:105618, 2020.

[53] Qi Liu, Shiwei Tong, Chuanren Liu, Hongke Zhao, Enhong Chen, Haiping Ma, and Shijin Wang. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 627–635, 2019.

[54] Paul S Adler and Kim B Clark. Behind the learning curve: A sketch of the learning process. *Management Science*, 37(3):267–281, 1991.

[55] Hermann Ebbinghaus. Memory: A contribution to experimental psychology. *Annals of neurosciences*, 20(4):155–156, 2013.

[56] Frank B Baker. *The basics of item response theory.* ERIC, 2001.

[57] Burrhus Frederic Skinner. Teaching machines. *Science*, 128(3330):969–977, 1958.

[58] Leonard Uhr. Teaching machine programs that generate problems as a function of interaction with students. In *Proceedings of the 1969 24th national conference*, pages 125–134, 1969.

[59] Jonathan D Wexler. Information networks in generative computer-assisted instruction. *IEEE Transactions on Man-Machine Systems*, 11(4):181–190, 1970.

[60] Jaime R Carbonell. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. *IEEE transactions on man-machine systems*, 11(4):190–202, 1970.

[61] Ali Alkhatlan and Jugal Kalita. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *arXiv preprint arXiv:1812.09628*, 2018.

[62] Jimmy De La Torre. Dina model and parameter estimation: A didactic. *Journal of educational and behavioral statistics*, 34(1):115–130, 2009.

[63] Ryan SJ d Baker, Albert T Corbett, and Vincent Aleven. More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing. In *International conference on intelligent tutoring systems*, pages 406–415. Springer, Berlin, Heidelberg, 2008.

[64] Nguyen Thai-Nghe and Lars Schmidt-Thieme. Multi-relational factorization models for student modeling in intelligent tutoring systems. In *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, pages 61–66. IEEE, 2015.

[65] Shaghayegh Sahebi, Yu-Ru Lin, and Peter Brusilovsky. Tensor factorization for student modeling and performance prediction in unstructured domain. *International Educational Data Mining Society*, 2016.

[66] Yuan Sun, Shiwei Ye, Shunya Inoue, and Yi Sun. Alternating recursive method for q-matrix learning. In *Proceedings of the 7th International Conference on Educational Data Mining*, pages 14–20, 2014.

[67] Neil D Fleming and Colleen Mills. Not another inventory, rather a catalyst for reflection. *To improve the academy*, 11(1):137–155, 1992.

[68] Richard M Felder, Linda K Silverman, et al. Learning and teaching styles in engineering education. *Engineering education*, 78(7):674–681, 1988.

[69] Hoang Tieu Binh, Nguyen Quang Trung, et al. Responsive student model in an intelligent tutoring system and its evaluation. *Education and Information Technologies*, pages 1–23, 2021.

[70] Zachary A Pardos, Ryan SJD Baker, Maria OCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of Learning Analytics*, 1(1):107–128, 2014.

[71] Ryan S Baker, Jaclyn Ocumpaugh, Sujith M Gowda, Amy M Kamarainen, and Shari J Metcalf. Extending log-based affect detection to a multi-user virtual environment for science. In *International Conference on user modeling, adaptation, and personalization*, pages 290–300. Springer, 2014.

[72] Abir Abyaa, Mohammed Khalidi Idrissi, and Samir Bennani. Learner modelling: systematic review of the literature from the last 5 years. *Educational Technology Research and Development*, 67(5):1105–1143, 2019.

[73] Robert V Lindsey, Mohammad Khajah, and Michael C Mozer. Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in neural information processing systems*, pages 1386–1394. Citeseer, 2014.

[74] Michel C Desmarais, Behzad Beheshti, and Rhouma Naceur. Item to skills mapping: deriving a conjunctive q-matrix from data. In *International Conference on Intelligent Tutoring Systems*, pages 454–463. Springer, 2012.

[75] Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, pages 1–27, 2020.

[76] Jingchen Liu, Gongjun Xu, and Zhiliang Ying. Data-driven learning of q-matrix. *Applied psychological measurement*, 36(7):548–564, 2012.

[77] Jiani Zhang and Irwin King. Topological order discovery via deep knowledge tracing. In *International Conference on Neural Information Processing*, pages 112–119. Springer, 2016.

[78] Penghe Chen, Yu Lu, Vincent W Zheng, Xiyang Chen, and Xiaoqing Li. An automatic knowledge graph construction system for k-12 education. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, pages 1–4, 2018.

[79] Wayne Xin Zhao, Wenhui Zhang, Yulan He, Xing Xie, and Ji-Rong Wen. Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Transactions on Information Systems (TOIS)*, 36(3):1–33, 2018.

[80] Radek Pelánek, Tomáš Effenberger, and Jaroslav Čechák. Complexity and difficulty of items in learning systems. *International Journal of Artificial Intelligence in Education*, pages 1–37, 2021.

[81] Zhengyang Wu, Ming Li, Yong Tang, and Qingyu Liang. Exercise recommendation based on knowledge concept prediction. *Knowledge-Based Systems*, 210:106481, 2020.

[82] Crystal Han-Huei Tsay, Alexander Kofinas, and Jing Luo. Enhancing student learning experience with technology-mediated gamification: An empirical study. *Computers & Education*, 121:1–17, 2018.

[83] Yujian Zhou, Reva Freedman, Michael Glass, Joel A Michael, Allen A Rovick, and Martha W Evens. Delivering hints in a dialogue-based intelligent tutoring system. In *AAAI/IAAI*, pages 128–134, 1999.

[84] Saiying Steenbergen-Hu and Harris Cooper. A meta-analysis of the effectiveness of intelligent tutoring systems on k–12 students' mathematical learning. *Journal of educational psychology*, 105(4):970, 2013.

[85] Saiying Steenbergen-Hu and Harris Cooper. A meta-analysis of the effectiveness of intelligent tutoring systems on college students' academic learning. *Journal of educational psychology*, 106(2):331, 2014.

[86] James A Kulik and JD Fletcher. Effectiveness of intelligent tutoring systems: a meta-analytic review. *Review of educational research*, 86(1):42–78, 2016.

[87] Wenting Ma, Olusola O Adesope, John C Nesbit, and Qing Liu. Intelligent tutoring systems and learning outcomes: A meta-analysis. *Journal of educational psychology*, 106(4):901, 2014.

[88] Konstantina Chrysafiadi and Maria Virvou. Student modeling approaches: A literature review for the last decade. *Expert Systems with Applications*, 40(11):4715–4729, 2013.

[89] Ma Mercedes T Rodrigo, Ryan SJd Baker, Maria CV Lagud, Sheryl Ann L Lim, Alexis F Macapanpan, SAMS Pascua, Jerry Q Santillano, Leima RS Sevilla, Jessica O Sugay, Sinath Tep, et al. Affect and usage choices in simulation problem solving environments. *Frontiers in Artificial Intelligence and Applications*, 158:145, 2007.

[90] Noboru Matsuda, Tadanobu Furukawa, Norman Bier, and Christos Faloutsos. Machine beats experts: Automatic discovery of skill models for data-driven online course refinement. *International Educational Data Mining Society*, 2015.

[91] Zachary A Pardos and Anant Dadu. dafm: Fusing psychometric and connectionist modeling for q-matrix refinement. *JEDM| Journal of Educational Data Mining*, 10(2):1–27, 2018.

[92] Michel Desmarais, Behzad Beheshti, and Peng Xu. The refinement of a q-matrix: Assessing methods to validate tasks to skills mapping. In *Educational Data Mining 2014*, 2014.

[93] Sein Minn, Feida Zhu, and Michel C Desmarais. Improving knowledge tracing model by integrating problem difficulty. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1505–1506. IEEE, 2018.

[94] Zichao Wang, Sebastian Tschiatschek, Simon Woodhead, José Miguel Hernández-Lobato, Simon Peyton Jones, Richard G Baraniuk, and Cheng Zhang. Educational question mining at scale: Prediction, analysis and personalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 15669–15677, 2021.

[95] Hao Cen, Kenneth Koedinger, and Brian Junker. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *Ikeda M., Ashley K.D., Chan TW. (eds) Intelligent Tutoring Systems, ITS 2006*, pages 164–175. Springer, Berlin, Heidelberg, 2006.

[96] Philip I Pavlik, Hao Cen, and Kenneth R Koedinger. Performance factors analysis–a new alternative to knowledge tracing. pages 531–538, 2009.

[97] Maciej Pankiewicz and Marcin Bator. On-the-fly estimation of task difficulty for item-based adaptive online learning environments. In *Proceedings of the 26th ACM Conference on Innovation and Technology in Computer Science Education V. 1*, pages 317–323, 2021.

[98] Kelly Wauters, Piet Desmet, and Wim Van Den Noortgate. Item difficulty estimation: An auspicious collaboration between data and judgment. *Computers & Education*, 58(4):1183–1193, 2012.

[99] Elizabeth Ayers and BW Junker. Do skills combine additively to predict task difficulty in eighth grade mathematics. Educational Data Mining: Papers from the AAAI Workshop, 2006.

[100] Luca Benedetto, Andrea Cappelli, Roberto Turrin, and Paolo Cremonesi. R2de: a nlp approach to estimating irt parameters of newly generated questions. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*, pages 412–421, 2020.

[101] Jiansheng Fang, Wei Zhao, and Dongya Jia. Exercise difficulty prediction in online education systems. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 311–317. IEEE, 2019.

[102] Radek Pelánek. Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies*, 13(2):354–366, 2019.

[103] Jirí Rihák and Radek Pelánek. Measuring similarity of educational items using data on learners' performance. *International Educational Data Mining Society*, 2017.

[104] Tanya Nazaretsky, Sara Hershkovitz, and Giora Alexandron. Kappa learning: A new method for measuring similarity between educational items using performance data. *arXiv preprint arXiv:1812.08390*, 2018.

[105] Dominic Mussack, Rory Flemming, Paul Schrater, and Pedro Cardoso-Leite. Towards discovering problem similarity through deep learning: Combining problem features and user behavior. In *Proceedings of The 12th International Conference on Educational Data Mining (EDM 2019)*, volume 615, page 618, 2019.

[106] Qi Liu, Zai Huang, Zhenya Huang, Chuanren Liu, Enhong Chen, Yu Su, and Guoping Hu. Finding similar exercises in online education systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1821–1830, 2018.

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

[108] Tanja Käser, Severin Klingler, Alexander G Schwing, and Markus Gross. Dynamic bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 10(4):450–462, 2017.

[109] Liangming Pan, Chengjiang Li, Juanzi Li, and Jie Tang. Prerequisite relation learning for concepts in moocs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1447–1456, 2017.

[110] Chun Wang and Jing Lu. Learning attribute hierarchies from data: Two exploratory approaches. *Journal of Educational and Behavioral Statistics*, 46(1):58–84, 2021.

[111] Mark D Reckase. Multidimensional item response theory models. In *Multidimensional item response theory*, pages 79–112. Springer, 2009.

[112] Louis V DiBello, Louis A Roussos, and William Stout. Review of cognitively diagnostic assessment and a summary of psychometric models. *Handbook of statistics*, 26:979–1030, 2006.

[113] Curtis Tatsuoka, Douglas H Clements, Julie Sarama, Andrew Izsák, Chandra Hawley Orrill, Jimmy de la Torre, K Tatsuoka, and Elvira Khasanova. Developing workable attributes for psychometric models based on the q-matrix. *Psychometric methods in mathematics education: Opportunities, challenges, and interdisciplinary collaborations*, 2017.

[114] Jimmy de la Torre and Nathan Minchen. Cognitively diagnostic assessments and the cognitive diagnosis model framework. *Educational Psychology*, 20(2):89–97, 2014.

[115] Mark D Reckase and Robert L McKinley. The discriminating power of items that measure more than one dimension. *Applied psychological measurement*, 15(4):361–373, 1991.

[116] Steffen Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[117] Zichao Wang, Yi Gu, Andrew Lan, and Richard Baraniuk. Varfa: A variational factor analysis framework for efficient bayesian learning analytics. *arXiv preprint arXiv:2005.13107*, 2020.

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

[119] Mohammad Khajah, Robert V Lindsey, and Michael C Mozer. How deep is knowledge tracing? In *Proceedings of the 9th International Conference on Educational Data Mining*, pages 94–101, 2016.

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

[121] Sein Minn, Yi Yu, Michel C Desmarais, Feida Zhu, and Jill-Jênn Vie. Deep knowledge tracing and dynamic student classification for knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1182–1187. IEEE, 2018.

[122] Chun-Kit Yeung. Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*, 2019.

[123] Heonseok Ha, Uiwon Hwang, Yongjun Hong, and Sungroh Yoon. Memory-augmented neural networks for knowledge tracing from the perspective of learning and forgetting. *arXiv preprint arXiv:1805.10768*, 2018.

[124] Fangzhe Ai, Yishuai Chen, Yuchun Guo, Yongxiang Zhao, Zhenzhu Wang, Guowei Fu, and Guangyan Wang. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In *Proceedings of The 12th International Conference on Educational Data Mining*, pages 240–245, 2019.

[125] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[126] Aritra Ghosh, Neil Heffernan, and Andrew S Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2330–2339, 2020.

[127] Shalini Pandey and Jaideep Srivastava. Rkt: Relation-aware self-attention for knowledge tracing. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1205–1214, 2020.

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

[129] Kevin H Wilson, Xiaolu Xiong, Mohammad Khajah, Robert V Lindsey, Siyuan Zhao, Yan Karklin, Eric G Van Inwegen, Bojian Han, Chaitanya Ekanadham, Joseph E Beck, et al. Estimating student proficiency: Deep learning is not the panacea. In *In 30th Conference on Neural Information Processing Systems, Workshop on Machine Learning for Education*, pages 1–8, 2016.

[130] Kevin H Wilson, Yan Karklin, Bojian Han, and Chaitanya Ekanadham. Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation. pages 539–544, 2016.

[131] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1848–1858, 2016.

[132] Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[133] Anna Saranti, Behnam Taraghi, Martin Ebner, and Andreas Holzinger. Insights into learning competence through probabilistic graphical models. In *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, pages 250–271. Springer, 2019.

[134] Zachary A Pardos and Neil T Heffernan. KT-IDEM: introducing item difficulty to the knowledge tracing model. In *International conference on user modeling, adaptation, and personalization*, pages 243–254. Springer, 2011.

[135] Xiaojing Wang, James O Berger, Donald S Burdick, et al. Bayesian analysis of dynamic item response models in educational testing. *The Annals of Applied Statistics*, 7(1):126–153, 2013.

[136] Stephen E Fancsali, Michael V Yudelson, Susan R Berman, and Steven Ritter. Intelligent instructional hand offs. *International Educational Data Mining Society*, 2018.

[137] Beverly Park Woolf. *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning.* Morgan Kaufmann, 2010.

[138] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[139] Vincent AWMM Aleven and Kenneth R Koedinger. An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor. *Cognitive science*, 26(2):147–179, 2002.

[140] Neil T Heffernan and Cristina Lindquist Heffernan. The assistments ecosystem: Building a platform that brings scientists and teachers together for minimally invasive research on human learning and teaching. *International Journal of Artificial Intelligence in Education*, 24(4):470–497, 2014.

[141] Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. Ednet: A large-scale hierarchical dataset in education. In *International Conference on Artificial Intelligence in Education*, pages 69–73. Springer, 2020.

[142] Xiaolu Xiong, Siyuan Zhao, Eric G Van Inwegen, and Joseph E Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.

[143] Jin Huang and Charles X Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on knowledge and Data Engineering*, 17(3):299–310, 2005.

[144] Wenbin Gan, Yuan Sun, Shiwei Ye, Ye Fan, and Yi Sun. Field-aware knowledge tracing machine by modelling students' dynamic learning procedure and item difficulty. In *2019 International Conference on Data Mining Workshops (ICDMW)*, pages 1045–1046. IEEE, 2019.

[145] Zhiwei Wang, Xiaoqin Feng, Jiliang Tang, Gale Yan Huang, and Zitao Liu. Deep knowledge tracing with side information. In *International Conference on Artificial Intelligence in Education*, pages 303–308. Springer, 2019.

[146] Kenneth Kotovsky and Herbert A Simon. What makes some problems really hard: Explorations in the problem space of difficulty. *Cognitive psychology*, 22(2):143–183, 1990.

[147] Klaus D Kubinger and Christian H Gottschall. Item difficulty of multiple choice tests dependant on different item response formats–an experiment in fundamental research on psychological assessment. *Psychology science*, 49(4):361–374, 2007.

[148] Gabriella Daroczy, Magdalena Wolska, Walt Detmar Meurers, and Hans-Christoph Nuerk. Word problems: a review of linguistic and numerical factors contributing to their difficulty. *Frontiers in psychology*, 6:348, 2015.

[149] Kenneth Kotovsky, John R Hayes, and Herbert A Simon. Why are some problems hard? evidence from tower of hanoi. *Cognitive psychology*, 17(2):248–294, 1985.

[150] Steffen Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology*, 3(3):57, 2012.

[151] Bernard L Welch. The generalization of "student's" problem when several different population variances are involved. *Biometrika*, 34(1/2):28–35, 1947.

[152] Wenbin Gan, Yuan Sun, and Yi Sun. Knowledge structure enhanced graph representation learning model for attentive knowledge tracing. *International Journal of Intelligent Systems*, 37(3):2012–2045, 2022.

[153] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 135–144, 2017.

[154] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[155] Tanya Nazaretsky, Sara Hershkovitz, and Giora Alexandron. Kappa learning: A new item-similarity method for clustering educational items from response data. *International Educational Data Mining Society*, 2019.

[156] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[157] Shanghui Yang, Mengxia Zhu, Jingyang Hou, and Xuesong Lu. Deep knowledge tracing with convolutions. *arXiv preprint arXiv:2008.01169*, 2020.

[158] Dongmin Shin, Yugeun Shim, Hangyeol Yu, Seewoo Lee, Byungsoo Kim, and Youngduck Choi. Saint+: Integrating temporal features for ednet correctness prediction. *arXiv preprint arXiv:2010.12042*, 2020.

[159] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[160] Wenbin Gan, Yuan Sun, and Yi Sun. Knowledge interaction enhanced sequential modeling for interpretable learner knowledge diagnosis in intelligent education systems. *Neurocomputing*, 2022.

[161] Kenneth R Koedinger, Emma Brunskill, Ryan SJd Baker, Elizabeth A McLaughlin, and John Stamper. New potentials for data-driven intelligent tutoring system development and optimization. *AI Magazine*, 34(3):27–41, 2013.

[162] Abhishek Singh Rathore and Siddhartha Kumar Arjaria. Intelligent tutoring system. In *Utilizing Educational Data Mining Techniques for Improved Learning: Emerging Research and Opportunities*, pages 121–144. IGI Global, 2020.

[163] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850, 2016.

[164] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[165] Fei Wang, Qi Liu, Enhong Chen, Zhenya Huang, Yuying Chen, Yu Yin, Zai Huang, and Shijin Wang. Neural cognitive diagnosis for intelligent education systems. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

[166] Song Cheng, Qi Liu, Enhong Chen, Zai Huang, Zhenya Huang, Yiying Chen, Haiping Ma, and Guoping Hu. Dirt: Deep learning enhanced item response theory for cognitive diagnosis. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2397–2400, 2019.

[167] Jan Schneider, Dirk Börner, Peter Van Rosmalen, and Marcus Specht. Augmenting the senses: a review on sensor-based learning support. *Sensors*, 15(2):4097–4133, 2015.

[168] Daniele Di Mitri, Jan Schneider, Marcus Specht, and Hendrik Drachsler. From signals to knowledge: A conceptual model for multimodal learning analytics. *Journal of Computer Assisted Learning*, 34(4):338–349, 2018.

[169] Albrecht Fortenbacher, Manuel Ninaus, Haeseon Yun, René Helbig, and Korbinian Moeller. Sensor based adaptive learning-lessons learned. *DELFI 2019*, 2019.

[170] Yutao Wang and Neil Heffernan. Extending knowledge tracing to allow partial credit: Using continuous versus binary nodes. In *International conference on artificial intelligence in education*, pages 181–188. Springer, 2013.

[171] Tomas Effenberger and Radek Pelánek. Validity and reliability of student models for problem-solving activities. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 1–11, 2021.

[172] Radek Pelánek and Tomáš Effenberger. Beyond binary correctness: Classification of students’ answers in learning systems. *User Modeling and User-Adapted Interaction*, 30(5):867–893, 2020.

[173] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[174] Zhenya Huang, Qi Liu, Chengxiang Zhai, Yu Yin, Enhong Chen, Weibo Gao, and Guoping Hu. Exploring multi-objective exercise recommendations in online education systems. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1261–1270, 2019.

A

List of Abbreviations

A-D

AI	artificial intelligence
AIED	artificial intelligence in education
AFM	additional factor model
ACT-R	Adaptive Control of Thought–Rational
AUC	area under the curve
ACC	prediction accuracy
BoW	bag-of-words
BKT	Bayesian knowledge tracing
CAI	computer assisted instruction
CT	Cognitive Tutor
CDA	cognitive diagnostic assessment
DINA	deterministic-input, noisy “and” gate
DLA	dynamic learner assessment
DKT	deep knowledge tracing

DKVMN dynamic key-value memory network

DAS3H item Difficulty, student Ability, Skill, and Student Skill practice History

DASH Difficulty, Ability, and Student History

DKT-DSC deep knowledge tracing with dynamic student classification

E-L

EDM educational data mining

FM factorization machine

FSLS Felder–Silverman learning style

GCN graph convolutional network

GNN graph neural network

HMM hidden Markov model

ITS intelligent tutoring systems

ICAI intelligent computer assisted instruction

IRT item response model

IoT internet of things

KT knowledge tracing

KC knowledge component

KTM knowledge tracing machine

KTM-DLF Knowledge Tracing Machine by modeling cognitive item Difficulty and Learning and Forgetting

KS knowledge structure

KSGKT KS–enhanced graph representation learning model for knowledge tracing

KIEDCDA knowledge interaction-enhanced dynamic cognitive diagnostic assessment

LSTM long short-term memory

M-V

MOOC massive open online course

MF matrix factorization

MIRT multidimensional item response theory

MANN memory-augmented neural network

NLP natural language processing

NLL negative log-likelihood

PFA performance factor analysis

QSQ question–skill–question

RNN recurrent neural network

SKVMN sequential key-value memory network

VARK Visual, Aural, Read/write, and Kinesthetic

VAE Variational Autoencoder