Deep Learning Based Knowledge Tracing: A Review, a Tool and Empirical Studies

Zitao Liu, *Member, IEEE*, Teng Guo, Qianru Liang, Mingliang Hou, Bojun Zhan, Jiliang Tang, Weiqi Luo, and Jian Weng

(Survey Paper)

***Abstract*—Knowledge tracing (KT) involves utilizing historical data from students' learning interactions to model their mastery of knowledge over time, with the aim of predicting their future performance in interactions. Recently, significant advancements have been achieved through the application of various deep learning methodologies to address the KT challenge. However, a considerable proportion of deep learning-based knowledge tracing (DLKT) approaches exhibit striking similarities in their methodologies, and model designs, and even the outcomes demonstrate minimal divergence. In addition, the evaluation procedures employed in current DLKT studies are not standardized, resulting in substantial inconsistencies in the reported area under the curve (AUC) outcomes, despite analyzing the same model on identical datasets. To address the two aforementioned problems, this paper proposes a generalized DLKT framework and represents the existing DLKT models with five components, i.e., multimodal data encoder, student knowledge memory, auxiliary knowledge base, learning outcome objective, and computational efficiency and scalability. Furthermore, we develop and open source a standardized DLKT benchmark platform named PYKT,[1] that consists of a standardized set of integrated data preprocessing procedures on 9 popular datasets across different domains, and 21 frequently compared DLKT model implementations. With PYKT, we conduct empirical and reproducible research to assess the performance of prevalent DLKT algorithms in an unbiased and clear setting over multiple data sources. Finally, we discuss the applications of KT techniques in the educational sector and their future development directions.**

***Index Terms*—Knowledge tracing, personalized learning, assessment, AI in education.**

I. Introduction

PERSONALIZED learning refers to an educational approach that tailors educational experiences to the unique needs, interests, and abilities of individual learners [1], [2], [3], [4]. Personalized learning is a broad educational paradigm, and knowledge tracing (KT) plays a critical role within this framework by providing the tools necessary for dynamically measuring and tracking students' knowledge states.

Essentially, KT is a sequential prediction task that models students' knowledge states from their historical interactions with platforms such as Massive Open Online Courses (MOOCs) and Intelligent Tutoring Systems (ITS). Fig. 1 presents a typical KT framework pipeline. For instance, consider a student, Jack, who has attempted five questions covering five knowledge components (KCs): multiplication, addition, division, subtraction, and equations. Each KC represents an independent cognitive structure or process utilized by learners to perform task steps or solve problems. We model Jack's knowledge states based on his responses to these questions, and changes in his knowledge states are illustrated through a radar chart at the bottom of the figure. The objective of KT is to model these states based on Jack's initial performance to predict his success in future interactions, such as correctly answering a subsequent division question, '$16 \div 4 =$'.

Corbett and Anderson, as early as 1994, pioneered the quantification of student knowledge at KC level [5]. Subsequent efforts expanded on this foundation, addressing the complexities of knowledge changing through probabilistic graphical models and factor analysis techniques. However, education operates as a dynamic and interconnected system, including a series of factors such as student's learning abilities, difficulties of questions and connections among KCs. These elements form a complex, evolving system that significantly affects the students' learning outcomes, challenging traditional machine learning algorithms based on statistical theory. Recently, with the swift progress of deep neural networks, models for KT based on deep learning (DLKT) have emerged as the predominant framework for analyzing students' understanding of KCs. DLKT models have achieved promising results due to their ability to capture nonlinear complex patterns, scalability to process large-scale data, and generalization capability for unseen data.

To provide a clear scope for this study, our research primarily focuses on digital learning platforms, including MOOCs and ITS, which generate extensive student interaction data. The datasets analyzed are selected based on their relevance to these platforms, encompassing both question-level and KC-level data.

Received 24 April 2024; revised 17 January 2025; accepted 25 February 2025. Date of publication 19 March 2025; date of current version 7 July 2025. This work was supported in part by National Key R&D Program of China, under Grant 2022YFC3303600 and in part by the Key Laboratory of Smart Education of Guangdong Higher Education Institutes, Jinan University under Grant 2022LSYS003. Recommended for acceptance by Z. Guan. *(Corresponding authors: Teng Guo; Qianru Liang.)*

Zitao Liu, Teng Guo, Qianru Liang, Bojun Zhan, Weiqi Luo, and Jian Weng are with the Jinan University, Guangzhou 510632, China (e-mail: liuzitao@jnu.edu.cn; tengguo@jnu.edu.cn; liangqr@jnu.edu.cn; zbj0613@stu2022.jnu.edu.cn; lwq@jnu.edu.cn; cryptjweng@gmail.com).

Mingliang Hou is with TAL Education Group, Beijing 100080, China (e-mail: houmingliang@tal.com).

Jiliang Tang is with the Data Science and Engineering Lab, Michigan State University, East Lansing, MI 48824 USA (e-mail: tangjili@msu.edu).

https://pykt.org/.

Digital Object Identifier 10.1109/TKDE.2025.3552759

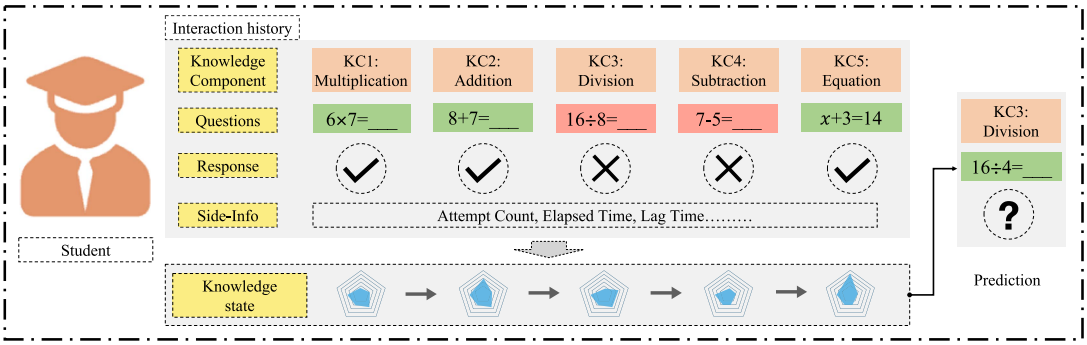

Fig. 1. The illustration of KT task.

This scope ensures that our findings could be applicable to personalized learning environments that require large scale data processing and the ability to model complex student learning behaviors. By defining these research boundaries, we aim to provide a more focused analysis of DLKT models in the context of personalized education.

Despite the innovations of DLKT in addressing KT challenges and achievingpromising outcomes [6], [7], [8], [9], [10], two primary issues continue to impede progress. First, a notable homogeneity exists in methodological approaches [8], [11], [12], necessitating a comprehensive review to discern trends and variations. Second, a lack of standardized evaluation in DLKT studies leads to inconsistent area under the curve (AUC) results, highlighting the need for benchmarking against an array of methods and a universal platform for fair comparison and real-world application assessment.

In recent years, many scholars have published review papers that analyze and summarize the current status and development trends in the field of KT. For example, Dai et al. [13] provided a comprehensive review of various KT models and techniques, assessing their effectiveness in predicting and understanding student learning behaviors. This survey classified the existing literature based on methodological approaches, including statistical, probabilistic, and deep learning models, highlighting how each contributes differently to the understanding and advancement of personalized education. Song et al. [14] provided a survey on the evolution and categorization of DLKT models, focusing on historical and technological advancements. They systematically classified the models into four primary categories: deep KT (DKT) and its variants, memory network-based KT, attention mechanism-based KT, and graph structural DLKT models. Abdelrahman et al. [15] conducted a review on KT, delineating the field's evolution from Bayesian inference and factor analysis to modern deep learning techniques. They classified KT approaches into traditional models and deep learning models, further subdividing the latter into sequence modeling, memory-augmented, attentive, and graph-based models. Although these surveys have provided detailed reviews of DLKT methods from various perspectives, they have not evaluated specific works. This leads to a problem: the factors contributing to the success of DLKT architectures in KT tasks and the performance of DLKT models in real-world educational contexts remain somewhat unknown. Therefore, there is a pressing need for a standardized DLKT benchmark platform to ensure methods can be compared fairly and transparently. Researchers need to be able to evaluate their proposed methods against a wide range of state of the art methods on both publicly available and private datasets, and practitioners need to be capable of discerning the advantages and disadvantages of DLKT algorithms in real-world educational contexts. To this end, Liu et al. [16] introduced an algorithm library named EduKTM, which contains a series of mainstream algorithmic models. However, EduKTM does not provide empirical studies based on its algorithm library to demonstrate its performance and effectiveness. Furthermore, due to differences in data preprocessing, training, and testing architectures among various KT studies, constructing an extensive and readily expandable algorithm library that covers the current mainstream KT algorithms is exceedingly challenging (EduKTM includes eight mainstream architectures). Therefore, we argue that for the current KT research community, conducting a thorough and detailed review of existing KT to facilitate readers' quick understanding of the current DLKT development trajectory, and constructing a high quality, standardized DLKT benchmark platform are two critically important issues.

Unlike existing DLKT surveys, we have not categorized the literature according to methods or technologies. Instead, we offer a novel perspective for examining the existing literature, presenting a unified approach to survey the current DLKT frameworks from a procedural standpoint. We observe that existing KT frameworks essentially comprise four parts: 1) data feature extraction or representation, 2) neural network framework design, 3) KT data or information, and 4) loss function design. Based on this, we propose a unified, conceptual framework in this survey, named GenKT (generalized KT), which encompasses the following components: 1) multimodal data representations, 2) student knowledge memory, 3) auxiliary knowledge base, and 4) learning outcome objective, and 5) computational efficiency and scalability. Note that we have added 'computational efficiency and scalability' so that the reader can better understand the efficiency of the various algorithms. GenKT not only lowers the threshold for understanding complex models but also opens new directions for in-depth research in KT. Moreover, we develop PYKT, an easy-to-use and end-to-end PyTorch benchmark library, to promote reproducibility and encourage future research in KT community. PYKT includes necessary data preprocessing for 9 mainstream KT datasets, standardized dataset splitting

processes, implementations of 21 mainstream DLKT models (updated in real-time), and also provides tailored real-world evaluation methods and protocols specifically for educational contexts. PYKT is designed to serve as a comprehensive resource for researchers and practitioners alike, aiming to streamline the process of developing, testing, and deploying DLKT models in various educational settings. Finally, based on the proposed PYKT, we conducted empirical standardized evaluations on the current mainstream DLKT models, yielding many insightful findings that deepen readers' understanding of the current KT frameworks.

Our main contributions are shown as follows:

- We propose a unified conceptual KT framework GenKT to review existing studies. Based on GenKT, we summarize the similarities and differences of current mainstream DLKT algorithms from five essential components. This unified perspective can help readers better understand the essence of current KT research.
- We develop PYKT, a comprehensive KT benchmark library that includes essential data preprocessing, standardized dataset splitting, cutting-edge DLKT implementations, and evaluation protocols tailored for educational contexts.
- We conduct standardized empirical studies on the current mainstream DLKT models to fairly and effectively compare their similarities and differences.

The remainder of this survey is organized as follows. Section II describes a series of concepts related to KT. Section III presents the formal definition of the KT task and a generalized modeling framework. Section IV reviews the current research based on the proposed framework. Section V describes the proposed tool named PYKT. Section VI gives empirical studies based on the proposed tool. Section VII introduces KT-based education application and implications. Section VIII discusses some potential future research directions. Finally, Section IX provides a summary of the paper.

II. Background

Educational measurement situations typically deal with unobservable, latent traits (e.g., intelligence or reading ability) that cannot be measured directly as height [17]. To this end, a family of latent trait measurement models has been developed to measure and predict individuals' latent traits (e.g., ability, skill, or knowledge) in various subject domains. Particularly, educational measurement models, such as item response theory (IRT) and cognitive diagnosis models (CDMs), are highly correlated to the birth of KT models.

A. Item Response Theory

Modeling students' responses to test items with dichotomously or polytomously scored data, IRT aims to examine students' latent abilities based on their probabilities of correctly answering the items. IRT models take into account not only test-taker characteristics (e.g., ability, motivation), but also item characteristics, such as item discrimination, difficulty, and guessing. Students' performance is based on a latent trait continuum, and students with higher latent abilities have higher probabilities of answering the question correctly. As a statistical modeling framework, IRT includes a variety of models that can be found in the literature, for instance, the Rasch model [18], the two- to four-parameter logistic model (2PL - 4PL model) for dichotomous data [19], and the graded model series for polytomous data [20], [21].

B. Cognitive Diagnosis Models

CDMs are a family of restricted latent class models that classify students with respect to the absence or presence of a set of fine-grained attributes measured by a given test based on the response data and the Q-matrix of the test [22]. Attributes in CDMs are defined as the skills or procedures that must be mastered by students to correctly answer an item and can indicate students' thinking processes. Q-matrix delineates the relationship between the items and the attributes [23], [24]. A number of CDMs can be found in the literature [25], [26]. Some CDMs have restricted assumptions on the interplay between attribute vector and item responses [27], [28], [29], whereas the general CDMs subsume specific CDMs [30], [31], such as the generalized deterministic inputs, noisy and gate model. Unlike IRT, students' performance is represented by a binary latent attribute vector in CDMs indicating whether or not the student has mastered each attribute.

C. Relations Between KT and IRT and CDMs

Although KT, IRT, and CDMs are methodologies used to understand and model student learning and ability in the context of educational assessments, they have several differences in their methodology and practice. First, methodologically, the main purpose of KT is to track the probabilities of students having learned a particular skill or knowledge over time, and thus it is a longitudinal methodology in nature. In contrast, both IRT and CDMs have been developed and used primarily for assessments that measure students' abilities at a single time point. Although IRT and CDMs have been extended for longitudinal assessment data, their longitudinal applications are relatively rare. Second, in practice, KT is often used in computer-based learning environments (e.g., intelligent tutoring systems, adaptive learning platforms) that allow real-time tracking of student learning, whereas IRT is widely used in large-scale assessments to measure and rank students' latent abilities, such as standardized tests (e.g., SAT) and CDMs are used in diagnostic and formative assessments to provide detailed information about students' strengths and weaknesses. In recent years, both IRT and CDMs have been integrated with adaptive testing algorithms to enable educational assessments in computer adaptive testing.

Despite the differences between these methodologies, it is noteworthy that the first KT models were inspired by and have used educational measurement models, particularly IRT or CDMs, as references. For instance, the Bayesian KT (BKT) model [5] is a probabilistic model that estimates the probability of a student knowing a particular KCs based on their responses to the items, and it referenced IRT models. BKT has been widely used as a benchmark for evaluating other KT models. Other models have also borrowed from IRT and have been studied

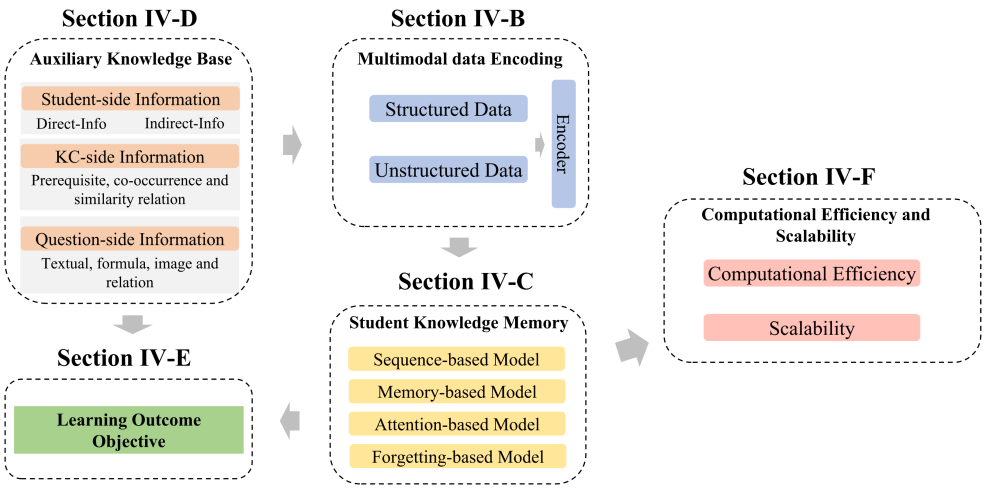

Fig. 2. The overview of the proposed GenKT modeling framework.

in this paper [32], [33], [34], [6], [35], [36]. On the other hand, the Performance Factors Analysis model is a KT model that uses latent class analysis to estimate the probability of a student knowing a particular skill or KCs, which is similar to CDMs because they both assume that students' responses to items are determined by a given set of latent skills or KCs [37]. Learning process-consistent KT (LPKT) is another example of a KT model that refers to CDMs and is investigated in this study [38]. Therefore, there is no doubt that the aforementioned IRT and CDMs models have strong relevance to the KT models that are the primary focus of this paper.

III. Problem Statement

Since the concept of KT was introduced in 1994 [5], a large number of scholars have made various improvements to related algorithms. However, the definition of KT has remained consistent over the years.

Given a student's historical interactions with exercises, represented as s in a set $|S|$, where $s = \{X_0, X_1, \ldots, X_t\}$ is a sequence of interactions $\{(e_0, a_0), (e_1, a_1), \ldots, (e_t, a_t)\}$ related to a specific concept or KC c from a set $|C|$. Specifically, $X_t = (e_t, a_t)$ represents the interaction at step t from a set T for an exercise e from a set $|E|$. The tuple indicates the correctness a_t of the result at step t, where a_t is a binary value: 1s indicating correct answer and 0s otherwise. The objective of KT is to predict the subsequent interaction X_{t+1} or the subsequent interaction sequence $X_{t+1} : X_{t+n}$, where n is the sequence length.

IV. The Generalized DLKT Framework

A. Framework Overview

To carefully assess the progress of DLKT model-wise and application-wise, we survey existing DLKT-related publications in top AI/ML venues[1] from 2015-2024.[2] After a comprehensive literature review, we present the following generalized KT framework, namely, GenKT, to unify the majority of recent KT studies that might seem to resemble each other with very limited nuances from the methodological perspective.

Specifically, our GenKT framework consists of five critical components: 1) multimodal data representations (See Section IV-B); 2) student knowledge memory (See Section IV-C); 3) auxiliary knowledge base (See Section IV-D); 4) learning outcome objective (Section IV-E); and 5) computational efficiency and scalability (Section IV-F). The overview of the proposed GenKT modeling framework is depicted in Fig. 2 and a comprehensive review of recent progress in each component is discussed in the following sections.

B. Multimodal Data Representation

In the domain of KT, the diversity of data types is a key characteristic, reflecting the multifaceted nature of student learning behaviors and educational contents. KT data can generally be categorized into structured data, such as question IDs, KC IDs, correctness labels, and temporal features like elapsed time, which provide insights into interaction dynamics; and unstructured data, including textual descriptions of questions, visual elements like images or formulas, and graph structures that model dependencies and interactions. Integrating these diverse data types enables KT models to comprehensively capture learning processes, leading to more accurate predictions and personalized educational interventions.

1) Structured Data Representation: In current KT studies, the representation of structured data primarily employs two methods: one-hot encoding [39], [40] and randomly generated trainable embeddings based on the uniform or normal distribution, as exemplified by the nn.Embedding() function in Pytorch [41].

In DLKT, DKT [8] initially adopts a multi-hot vector method to represent both questions and responses by concatenating their respective one-hot vectors. From a feature interaction perspective, this combined representation enables the neural

[1]Venues include NeurIPS, ICML, ICLR, AAAI, IJCAI, KDD, WWW, SIGIR, MM, WSDM, ICDM, CIKM.

[2]The very first deep knowledge tracing model is proposed by Piech et al. at NIPS 2015 [8].

network's internal weight matrix to directly model the associations between questions and responses, offering greater efficacy than encoding questions and responses separately via one-hot. Moreover, DKT explicitly indicates that for a large number of questions, it shifts to randomly generating dense embeddings for the representations of questions and responses.

Influenced by DKT's representation techniques, many subsequent KT studies have adopted the multi-hot encoding approach for data representations. Some studies utilize the multi-hot vector to represent the question and response [11], [42], [43], [33], [44], [45], and some studies generate the trainable embeddings for the question and response [46], [47], [48], [49], [50].

In addition to question and response information, some studies also have devised different fusion representation strategies for structured data based on their input information and architecture design. For example, SAINT+ utilizes continuous embedding and categorical embedding to represent the elapsed time and lag time. Pu et al. [46] added the lag time as the bias to the attention score computing as follows:

$$A_{i,j} = \frac{q_i k_j + b(\Delta_{i-j})}{\sqrt{d_k}}, \forall j \leq i, \tag{1}$$

where $b(\Delta_{i-j})$ is the time gap bias. ClickstreamKT [51] concatenates the categorical and continuous features together in their click-stream event. In multi-factors aware dual-attentional KT (MF-DAKT) [52], the recent factor is designed to record the attempts of students on relevant concepts of the questions. The recent factor is a multi-hot vector, whose dimension is three times the total number of KCs. $\mathcal{F}(\Delta_{u_i,c_k,T})$ is a forgetting function, where $\Delta_{u_i,c_k,T}$ represents the lag time of KC c_k on a question u_i, which is similar to repeated time gap in DKT-F [53].

Liu. et al. [54] constructed attribute features for each question, incorporating aspects related to question difficulty such as average response time and question type. For the -th question, its attribute features are denoted as $\mathbf{f}_i = [\mathbf{f}_{i1}; \ldots, \mathbf{f}_{im}]$, where m is the number of features. If the j-th feature is categorical (e.g., question type), then $\mathbf{f}_{ij}$ is a one-hot vector. If the j-th feature is numerical (e.g., average response time), then $\mathbf{f}_{ij}$ is a scalar value. Individual estimation KT (IEKT) [55] introduces cognition estimation (CE) to represent students' cognition levels on questions. CE consists of students' knowledge states and the question representations. The knowledge states are represented as the hidden states of the recurrent neural network(RNN) model. The question is represented as follows:

$$\mathbf{v}_i = \mathbf{e}_i^q \oplus \mathbf{e}_i^{-c}, \tag{2}$$

where $\mathbf{e}_i^q$ is the random embeddings of the i-th question and $\mathbf{e}_i^{-c}$ is the averaged embedding of the concepts which are related to the i-th question.

In KT, the representation of structured data involves the following patterns: first, due to the nature of label data (response as a binary sequence with only 0s and 1s), the use of one-hot encoding is quite common in current KT studies. Second, all information related to KT labels, such as time-related, KC-related, and forgetting-related information, is typically represented jointly. Third, when the data samples are small, multi-hot representations are often preferred; during this process, some numerical data may also be represented using one-hot encoding, as seen in SAINT+ for lag time and elapsed time representations. When the data samples are large, random vector embeddings are often used to represent multiple related features, as in IEKT [55].

2) Unstructured Data Representation: In this section, we provide a detailed description of the representation of unstructured data.

For text data, the most commonly used representation techniques often derive from representation frameworks commonly used in natural language processing (NLP), such as word2vec [56], bidirectional encoder representations from Transformers (BERT) [57], etc. Specifically, exercise-enhanced RNN (EERNN) [58] utilizes word2vec to represent the textual descriptions of questions and then input question embeddings into bi-directional long short-term memory (BiLSTM) [59] to learn the more fine-grained semantic correlations of questions. EERNN concatenate the learned question embeddings with the multi-hot question vectors. Adaptable KT (AdaptKT) [60] devises an unsupervised auto-encoder to incorporate the semantics of the texts of questions. The representation design of text data in AdaptKT is similar to EERNN. Differently, qDKT [33] constructes a multi-hot vector for a given question-response pair, treating it as a word in a natural language corpus. Subsequently, qDKT generates the embeddings for each word using the word embedding algorithm fastText [61]. Exercise hierarchical feature enhanced KT (EHFKT) [62] initially employs BERT to generate embeddings for questions. It then inputs these question embeddings into three distinct modules to produce separate embeddings, representing knowledge distribution, semantic features, and question difficulty, respectively. Ultimately, EHFKT concatenates all embedding vectors and inputs the combined result into a sequence model.

In addition to text data, some KT studies represent images and formula information. For instance, QuesNet [64] utilizes a random embedding layer, a convolutional neural network (CNN) layer, and a fully connected layer to convert text data, image data, and metadata into three distinct embeddings. Similar to EERNN, QuesNet also employs a BiLSTM to integrate these three embeddings. KT model that integrates mathematical exercise representation and association of exercise (ERAKT) [65] initially pre-processes LaTeX-represented formulas in questions via an ontology replacement method. It then employs BERT to generate embeddings for the questions.

Moreover, existing KT data hardly contains naturally-formed graph data and almost all graph data in current KT studies is predefined manually. For the representation of graph data, existing graph neural network (GNN) architectures are essentially employed. Modeling KT entities (e.g., KCs and questions) through graphs and representing nodes with GNNs offers an advantage - it allows the nodes' vector representations to capture and learn the high-order relational features among nodes. For example, deep hierarchical KT [66] constructs a question-question graph and used the LINE [67] and node2Vec [68] models to learn the representations of nodes. Graph based KT (GKT) [43] constructs a graph with KCs as nodes, where the relationships between nodes are the dependency relations of KCs. Then, GKT

utilizes the message-passing aggregation mechanism from GNN to generate representations for the nodes. Graph based interaction model for KT (GIKT) [50] captures the relationship between KCs and questions as a bipartite graph and then employed a GCN architecture to extract features from the nodes on the graph. To represent the fine-grained relationships between questions, hierarchical graph KT (HGKT) [69] constructs a hierarchical graph consisting of two layers: the bottom layer is a graph with questions as nodes (similar to the graph in DHKT), while the top layer is a graph with question categories as nodes. This hierarchical graph can represent associations between both individual questions and question categories, as well as establish hierarchical connections between questions and their categories.

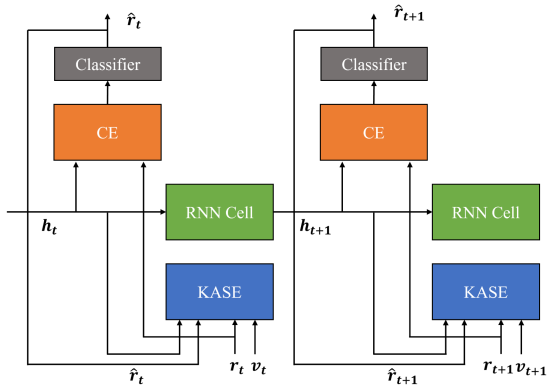

Fig. 3. The framework of IEKT. KASE is the knowledge acquisition sensitivity estimation module. CE is the cognition estimation module.

C. Student Knowledge Memory

The learning trajectory encompasses a student's progression from initial exposure to new information to achieving full mastery. Within this continuum, the student's knowledge memory is exceptionally intricate, evolving in response to diverse changes in time, environment, and tasks. Consequently, accurately modeling the dynamic nature of student knowledge memory is crucial for the efficacy of KT tasks.

Motivated by the significant advancements in deep learning [70], [71], recent studies in KT have increasingly adopted sophisticated neural network methodologies [8], [53], [66], [72], [73], [74], [75]. Simultaneously, with a detailed understanding of the learning process, researchers have endeavored to simulate the evolution of student knowledge memory from multiple perspectives. Based on various foundational principles, related studies can be systematically categorized into four types: 1) sequence-based modeling, 2) memory-based modeling, and 3) attention-based modeling. As KT fundamentally models human learning behavior, the concept of 'forgetting' has perennially accompanied KT studies [76], [77]. Therefore, in this subsection, we will also review frameworks related to forgetting as a distinct category, namely 4) forgetting-based modeling .

1) Sequence-Based Student Knowledge Memory Modeling: In 2015, Piech et al. [8] pioneered the use of neural network models to address KT tasks. Specifically, they employed an RNN, a deep learning model adept at processing sequential data, to delineate the evolution of student knowledge memory. Since this seminal study, a multitude of studies have rigorously explored KT tasks using sequence models such as RNN and long short term memory network (LSTM) [53], [66], [74], [78], furthering the domain's understanding and capabilities.

First, scholars have sought to ameliorate certain inherent limitations of the DKT model, particularly concerning interpretability, robustness, and overfitting. The DLKT model as referenced in [79], focuses on augmenting the interpretability of the fundamental RNN-based DKT model. It advocates for the application of layer-wise relevance propagation (LRP) to demystify the predictions made by DLKT models. Adversarial training based KT (ATKT), as proposed in [47], introduces an efficient adversarial training approach to address overfitting and fortify model robustness. This method strategically injects perturbations into the original interaction embedding to intensify the training challenge, thereby enhancing the model's generalization ability. Beyond adversarial techniques, ATKT also integrates LSTM with a knowledge hidden state attention module, sharpening the model's focus on the knowledge states pertinent to the current question and leading to more accurate predictions.

Second, several scholars have incorporated individual learning characteristics of students into the original DKT model. Minn et al. [80] introduced an enhancement to DKT, termed deep KT with dynamic student classification (DKT-DSC). DKT-DSC employs K-means clustering to categorize student profiles into groups based on their performance across KCs. Additionally, it dynamically updates the cluster information over time to reflect changes in student performance. IEKT [55] utilizes an LSTM as the foundational model and, building upon that, introduced a cognitive estimation module and a knowledge acquisition sensitivity estimation module to more accurately capture each student's unique cognitive abilities and knowledge acquisition capacities, as shown in Fig. 3.

Lastly, some scholars sought to further capture students' historical interaction patterns to enhance the performance of the original DKT model. GIKT [50] designs an interactive prediction approach based on the LSTM model. In addition to considering the interaction between the student's current state and the target question, it also accounts for the interactions between student's state and skills related to the question, as well as the interaction between historical practice and the target question and its associated KCs. This comprehensive approach enable the model to more thoroughly capture the student's knowledge state and the intricate relationships between questions and KCs. Su et al. [58] introduced EERNN model, a text-aware KT model designed to predict the likelihood of a student correctly answering a specific question. This model employs a BiLSTM module to derive a representation (i.e., a vector) of each question from its text, and then traces a student's knowledge states by integrating these representations with those of previously answered questions using another LSTM module. This methodology facilitates a nuanced understanding of knowledge acquisition and assessment dynamics. Knowledge query network (KQN) [11] designs a knowledge encoder and a KC encoder based on the

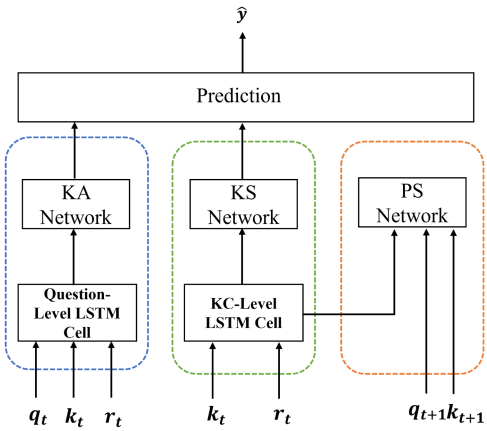

Fig. 4. The framework of QIKT. The blue dotted box represents the KA module, the green dotted box represents the KS module, and the orange dotted box represents the PS module.

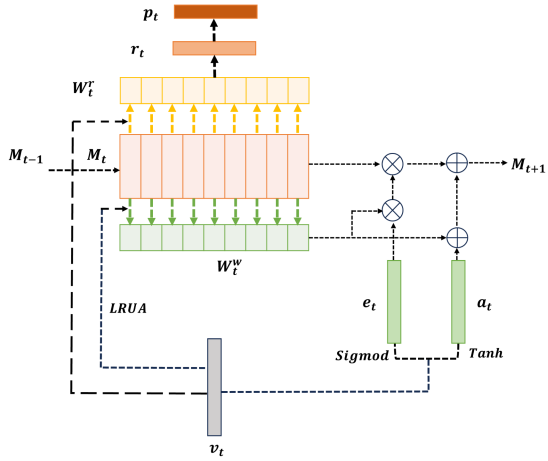

Fig. 5. The architecture of KVMN.

RNN model, embedding the student's knowledge state and the forthcoming questions, respectively. The interaction between the knowledge state vector and the skill vector was defined as the dot product of these two vectors. This interaction mechanism provided the model with a robust means to articulate the relationship between knowledge and skills, enhancing the accuracy of students' performance prediction. Question-centric interpretable KT (QIKT) [81] includes three modules, as shown in Fig. 4. The question-centric knowledge acquisition (KA) module assessesstudents' knowledge accumulation in response to specific questions over time, while the question agnostic knowledge state (KS) module captures the overall dynamics of knowledge states. Finally, the question-centric problem-solving (PS) module estimates the capabilities of students to tackle a specific question with their current knowledge states. The bifurcated design in QIKT enables a more nuanced understanding of how students interact with and retain educational content.

Sequence-based models, such as RNNs and LSTMs, are particularly strong in capturing the temporal dynamics of students' learning processes, effectively modeling how knowledge evolves over time. This approach offers enhanced interpretability, especially when mechanisms like LRP are employed. However, these models often encounter challenges of computational complexity, particularly with long sequences, as their sequential nature hinders parallelization. Additionally, they are prone to overfitting, especially in the absence of regularization techniques or adversarial training.

2) Memory-Based Student Knowledge Memory Modeling: In order to map the intricate KC acquired by learners, numerous studies expanded upon DKT by incorporating an additional memory structure. Specifically, in alignment with the key-value memory network (KVMN) [82], a key-value memory is utilized to depict the state of knowledge, offering a more robust representational capacity compared to the hidden variable in DKT. This key-value memory comprises two matrices, a key matrix holding the representations of KCs, and a value matrix recording the student's proficiency level for each KC.

Zhang et al. [10] proposed the dynamic key-value memory network (DKVMN), a version of DKT enhanced with two memory matrices: key and value (shown in Fig. 5). To track the progression of a student's knowledge state over time, DKVMN configures the value matrix to be dynamic, while maintaining the key matrix as static, thus facilitating a nuanced tracking of knowledge acquisition and application. Since then, a series of variants based on the KVMN structure have been proposed to enhance the performance of KT tasks. Abdelrahman et al. [83] proposed the sequential KVMN (SKVMN), designed to address a limitation in DKT and DKVMN, where the KCs needed to answer past questions in a sequence are not necessarily pertinent to the KCs needed to answer the current question. Therefore, SKVMN employs a modified LSTM for sequential modeling, termed Hop-LSTM, while maintaining the same key-value memory structure and loss function as in DKVMN. Meanwhile, Hop-LSTM can explicitly capture sequential dependencies among questions in a sequence of interactions, and update the student's knowledge state based on their responses to relevant questions.

Ai et al. [84] proposed the concept-aware dynamic key-value memory networks (DKVMN) model. They designed KC weights based on a list of KCs and reconstructed the original DKVMN model's memory structure, reading process, and updating process based on these improvements. This approach provides a more tailored and concept-focused structure to the dynamic modeling of students' knowledge states, enhancing the precision and adaptability of KT tasks. Yeung et al. [34] proposed an interpretable DKVMN model grounded in IRT. The principal concept posits that the probability of a student correctly answering a question depends on their learning ability and the question's difficulty. Attentive neural turing machine [85], building upon the neural turing machine (NTM) model, introduces an attention-based controller, differential read-write processes, and a specifically designed memory bank, making it more suitable for KT tasks. These modifications aim to enhance the model's performance in cold-start scenarios, especially in the context of limited learner data availability.

Memory-based models provide a comprehensive and adaptive framework for representing and updating students' knowledge states, making them highly effective in tracking mastery of

various KCs. They dynamically adjust to changes in students' learning progress, which is crucial for accurate modeling. However, the requirement to maintain and update memory units for each KC can lead to significant scalability and computational challenges, particularly in large-scale datasets. Moreover, the complexity of implementing additional memory structures can complicate the model architecture, making it more difficult in optimization.

3) Attention-Based Student Knowledge Memory Modeling: With the introduction of the Transformer architecture [86], it has been widely applied in various fields such as machine translation, traffic flow prediction, etc. Due to the model's use of attention mechanisms to process sequence data, it can capture long-term dependencies more effectively, In this case, several studies have endeavoured to integrate attention mechanisms into KT models. In 2019, Pandey et al. [7] was the first to apply the attention mechanism to KT tasks and proposed a model termed self-attentive KT (SAKT). The model employs the scaled dot-product attention mechanism to learn attention matrices using multiple attention heads [86]. Specifically, each attention matrix holds relative weights from a representative subspace, signifying the importance of past interaction questions in predicting a student's response to the current question. Subsequently, attention matrices from different representative subspaces are combined and forwarded to a feed-forward network to predict student performance.

Since then, a series of scholars have improved models based on the attention mechanism from various perspectives to enhance the performance of KT tasks. Pandey et al. [87] considered the relation between questions and students' forgetting behavior based on an attention-based model. They proposed an relation-aware self-attention model for KT (RKT) model with a novel attention mechanism that introduced a relation coefficient based on the original self-attention. This mechanism allows the model to pay more attention to elements strongly related to the target element when calculating attention weights.

Choi et al. [48] introduced a separated self-attentive neural KT (SAINT) model, which replaces self-attention with the Transformer architecture. SAINT incorporates an encoder-decoder model along with the scaled dot-product attention mechanism. Specifically, SAINT divides a student's sequence of interactions into question embedding and response embedding sequences, which are then input into the encoder and decoder, respectively. Both the encoder and decoder combines multi-head attention networks with the scaled dot-product attention mechanism. Shin et al. [49] introduced the SAINT+ model, which integrates two time-related features into SAINT, the time taken to answer each question and the time gap between two consecutive learning interactions, to improve the performance of the SAINT model.

Subsequently, a series of scholars made improvements based on the SAINT+ algorithm. Zhou et al. [88] proposed a novel model named levelled attentive KT (LANA) that considers the unique characteristics of KT tasks. They initially introduced modifications to the basic Transformer model, such as directly feeding positional embeddings into the attention modules. Furthermore, the LANA model employs a novel student-related feature extractor (SRFE) to abstract essential student-related features from the input sequence. Additionally, the LANA model utilizes a pivot module and the extracted student-related features to dynamically construct different decoders tailored to individual students. Ghosh et al. [6] introduced a decline in the weight of attention and proposed a novel model termed attentive KT (AKT). The attention mechanism in AKT, referred to as monotonic attention, utilizes an exponential decay curve to diminish the significance of exercises from the distant past. This exponential weight decay is designed to account for the diminishing retention or forgetting effect in a student's memory as time progresses, thereby aligning the model more closely with the observed patterns of memory retention and attrition.

Long et al. [89] introduced a new model named collaborative KT, inspired by the collaborative filtering techniques found in recommendation systems. They employed a novel attention mechanism termed collaborative multi-head self-attention to project the embedding vectors representing the knowledge states of similar peers. Concurrently, the model embeds a student's previous answer history to represent the student's knowledge state based on their answer history using the multi-head self-attention mechanism. Ultimately, these internal and external representations are combined to predict whether a student would answer the next question correctly. MF-DAKT [52] focuses on question-related information, such as the difficulty of questions, the relationships between questions, and students' most recent attempts on relevant concepts of the target question. To distinguish the contributions of individual factors and their interactions in different records, it introduces a dual-attentional KT prediction method named DAKT. This model employs two sub-spaces in conjunction with an attention mechanism to capture the information embedded in factors and their interactions from distinct viewpoints, offering a more nuanced understanding of the dynamics influencing student performance. Liu et al. [35] designed a robust yet straightforward baseline method for addressing the KT task, termed SimpleKT. SimpleKT streamlines the complex student knowledge state estimation component by utilizing the standard dot-product attention function, thus offering an efficient and effective approach to modeling students' knowledge acquisition and application processes.

Beyond self-attention and Transformer mechanisms, some studies have utilized the dot product [9] or cosine similarity [12] to implement attention mechanisms. Shen et al. [9] introduced the convolutional KT (CKT) model, which integrates strategies similar to dot-product attention mechanisms with 1-D convolutional networks to predict correct answers. The CKT model not only consideres the past question-answering sequence but also accounts for a student's individualized KCs, represented as both historically relevant performance and overall concept performance. Liu et al. [12] introduced an explainable exercise-aware KT (EKT) framework designed to monitor student progress across multiple distinct concepts simultaneously. They utilized cosine similarities to compute attention scores, thereby quantifying the relevance between new exercises and historical exercises.

Attention-based models, particularly those utilizing the Transformer architecture, excel in scalability due to their ability to process sequences in parallel, making them well-suited for

large datasets. They are also highly effective in capturing long-term dependencies in learning sequences, providing a more comprehensive understanding of students' knowledge states over time. Nonetheless, these models require substantial memory resources due to the use of multi-head attention mechanisms, and their computational complexity increases quadratically with sequence length, which can pose challenges in terms of efficiency and scalability.

4) Forgetting-Based Student Knowledge Memory Modeling: Forgetting constitutes a significant characteristic of the learning process and markedly impacts students' ultimate mastery of knowledge. Some researchers attempted to model student knowledge memory while incorporating the characteristics of forgetting, recognizing its integral role in the retention and recall of learned information. This approach seek to align the models more closely with the actual patterns of knowledge retention and attrition observed in learners.

Some scholars consider forgetting features from the perspective of learning progress. For instance, Shen et al. [38] introduced the LPKT model, which accounts for a student's learning gain during answer prediction. Learning gain is defined as the change in knowledge state over a time interval since the last answered the question, integrating the time lapse between answering questions into the embedding representations of the exercise sequence. The LPKT model comprises three sequential memory cells: 1) a learning progress gate that projects an embedding for the most recent exercise in the sequence, considering its tag, time to answer, and time-lapse before answering, 2) a forgetting gate that utilizes the latest two learning progress embeddings to project an output representing the forgetting effect, and 3) a prediction cell that combines the forgetting output and the latest question tag embedding to predict the correct answer.

Drawing inspiration from the Hawkes process [90], Wang et al. [91] introduced HawkesKT, a model employing point processes to dynamically model temporal cross-effects in KT tasks. The model posits that a student's mastery of a KC is influenced not only by prior interactions with questions related to the same KC but also by interactions with questions about other KCs, referred to as cross-effects. Additionally, the model suggests that the cross-effects stemming from different previous interactions could have varied temporal impacts on the mastery of different KCs. While all cross-effects diminish over time, their rates of decay are not uniform, indicating that certain KCs might be more prone to being forgotten compared to others.

Abdelrahman et al. [92] introduced the deep graph memory network (DGMN), a model merging graph neural networks with memory components to facilitate forgetting-aware KT. The model is designed to capture the forgetting behavior across a KC space, thereby identifying indirect relationships between questions. DGMN constructs a dynamic graph from knowledge state memory to discern relationships among KCs. Given a sequence of interactions, DGMN employs an attention mechanism to link questions to their pertinent KCs. Subsequently, it computes forgetting features over the sequence and integrated question embedding, KC graph embedding, and forgetting features through a gating mechanism. The output of this gating process is then used to estimate the probability of correctly answering subsequent questions.

Note that studies on KT considering forgetting behavior, such as [53], are not included in this section because they focus on features related to forgetting rather than modeling the student knowledge memory in the context of forgetting behavior. These studies prioritize the exploration of forgetting as a distinct phenomenon impacting learning, thereby diverging from the central theme of this discussion, which centers on the integration of forgetting into the modeling of student knowledge memory.

Forgetting-based models align well with human memory processes by incorporating time decay and other realistic features, leading to more accurate predictions of student performance over time. These models can be highly efficient, especially when employing simple forgetting mechanisms, making them suitable for large-scale applications. However, the simplicity of these mechanisms can sometimes oversimplify the learning process, potentially overlooking more complex patterns in student behavior. When more complex forgetting mechanisms are used, such as those involving LSTMs, scalability can become a significant challenge, particularly in large-scale settings.

D. Auxiliary Knowledge Base

As mentioned in the Section IV-B, the auxiliary knowledge base in KT research has become increasingly abundant. Currently, the auxiliary knowledge base essentially encompasses three types of information: student-side information, KC-side information, and question-side information.

1) Student-Side Information: Student-side information refers to data that is specific to each learner. It may include learning behaviors and habits, historical performance data, individual learning styles, and other variables that could impact the individual's learning process. It is crucial for the assessment of student progress and performance over time. Based on the method of collection and the nature of the content it reflects, student-side information in the auxiliary knowledge base is categorized into explicit student-side information and inferred student-side information.

Explicit student-side information refers to the information directly included in KT datasets about students. It encompasses the data fields that are explicitly available and recorded about the students' activities, characteristics, and responses within the online learning environment. Here, we list some commonly used direct student-side information in existing studies, as shown in Table I (Explicit Student-side Information). Note that lag time, elapsed time, and first response time all provide researchers with important temporal information about student learning behaviors, but they have distinct differences. The lag time focuses on the time interval between two consecutive tasks. By analyzing lag time, researchers can gain a better understanding of a student's learning pace, study habits, and potential moments of distraction or deep contemplation. Elapsed time focuses on the time it takes to complete a single task. It provides insights about the speed and depth at which a student works on a specific task or problem. On the other hand, first response time measures the time from when a student first sees a question to when they

TABLE I
SUMMARY OF THE STUDENT-SIDE INFORMATION

	Categorization	Information Title	Information Description
Student-side Information	Explicit Student-side Information	Correctness	Correctness is represented as a binary value where1 indicates correctness and 0 represents an incorrect response.
		First response time	First response time refers to the initial response time of a student to a particular question.
		Attempt count	Attempt count refers to the number of tries required to correctly answer each question.
		First action	First action represents the first operation of a student within the system. Specifically, it indicates whether the student initially attempts to answer the question or requests help first
		Elapsed time	Elapsed time is the time taken by a student to answer each question.
		Lag time	Lag time refers to the time interval between two consecutive learning interactions.
		Clickstream	Clickstream encompasses a comprehensive set of student-side information. It includes student ID, question ID, question's dimension and unit, event name,mouse location, timestamp, answer state, and the score corresponding to that answer state.
	Inferred Student-side Information (DKT-F)	Repeated time gap	Repeated time gap refers to the lag time between two interactions involving the same KC id.
		Sequence time gap	Sequence time gap represents the lag time between an interaction and the previous interaction in the sequence, regardless of whether these two interactions involve the same KC id.
		Past trial counts	Past trial counts refers to the frequency with which a student attempts questions with the same KC id.
	Inferred Student-side Information (CKT)	Historical relevant performance	Historical relevant performance indicates the relevant coefficients between present answering question and previous answered questions.
		Concept-wised percent correct	Concept-wised percent correct represents the student's comprehensive mastery of all KCs, calculated based on the percentage of correct answers the student has for each KC.
		Learning rate	Learning rate reflects the pace at which a student grasps KC.

provide their initial answer. This metric can help researchers discern the reaction speed of students when they first encounter a question, as well as the potential thought process or strategy selection they might undertake before attempting an answer.

The aforementioned direct student information is commonly used in KT tasks. Zhang et al. [93] argued that the DKT model holds significant potential. However, currently considering question (or KC) and correctness as inputs restricts the full potential of the DKT model. In their research, they adeptly incorporated student-side information such as first response time, attempt count, and first action, thereby significantly augmenting the effectiveness of their designed framework in KT tasks. Pu et al. [46] introduced the elapsed time into their KT framework. Subsequently, SAINT [48] further expands on this concept, incorporating not only the elapsed time but also adding the lag time to enhance the performance of KT frameworks. Besides, ClickstreamKT [51] argues that in the current KT tasks, researchers only consider the answers of students, and such coarse-grained information is not sufficient to capture all the details of a student's problem-solving process. Therefore, this model introduces clickstream data to capture the fine-grained activity information of students while they are solving problems. Overall, explicit student-side information has received widespread attention in the existing KT studies and has been proven to effectively enhance the performances of KT frameworks. Inferred student-side information refers to additional student-related data fields that researchers design and derive from explicit student-side information in KT datasets. This information is formulated through the analysis and interpretation of explicit data, aimed at quantifying deeper characteristics and attributes of students' learning behaviors that are not directly recorded or observable in the dataset. DKT-F [53] designs three derived quantitative indicators, including repeated time gap, sequence time gap, and past trial counts, based on explicit student-side information. These indicators aim to more deeply quantify students' forgetting behavior, demonstrating the application of inferred student-side information in practical research. We summarize these three indicators, as shown in Table I (Inferred Student-side Information (DKT-F)). Note that sequence time gap is similar to the previously mentioned lag Time. The reason for redefining it here is that the DKT-F aims to contrast it with the repeated time gap and to quantify the forgetting behavior of students during the learning process, as shown in Fig. 6.

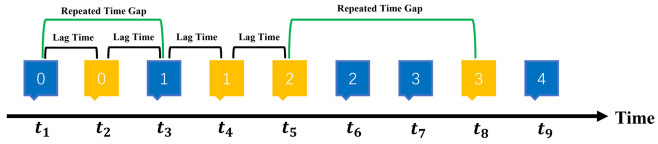

Fig. 6. An illustrative figure of a portion of student-side information, where blue and yellow represent distinct KCs. The numbers denote past trial counts, and the term 'lag time' essentially equals to the 'sequence time gap'.

DKT-DST [80] proposess constructing a learning ability profile to differentiate students' learning abilities. This profile is defined as a $1 \times N$ vector (where N represents the total number of KCs) intended to indicate the differences between a student's correct and incorrect performance on each KC. CKT [9] argues that a student's individualized prior knowledge and learning rate have significant implications for one's learning performance. CKT infers three student-side information, as shown in Table I (Inferred Student-side Information (CKT)). Compared to previous studies, MF-DAKT [52] introduces a novel inferred student-side information called recent factor, aimed at quantifying students' most recent attempts on relevant concepts to accentuate the impact of recent questions.

In the realm of student-side information, researchers have consistently attempted to profile students' learning behaviors and characteristics based on the raw system log data, aiming to enhance the performance of KT tasks.

2) KC-Side Information: KCs are interconnected, forming a variety of relationship types. Researchers have dedicated efforts to capturing the side information of KCs, primarily focusing on these interrelationships. Given the obscure nature of relationships among KCs, numerous methods have been employed to explore these connections. In this survey, we summarize several typical types of KC relationships identified in existing studies:1) Prerequisite relation indicates that certain concepts must be understood before others. If a student has mastered a

specific concept, it is presumed that they have also mastered the associated prerequisite concepts. 2) Co-occurrence relation represents that questions related to these two KCs often appear consecutively. 3) Similarity relation refers to KCs that belong to the same category.

Some studies utilize the pre-defined graph to model the relations of KCs. For example, prerequisite-driven deep KT with constraint modeling (PDKT-C) [94] uses the homogeneous unweighted directed graph to represent the prerequisite relations (prerequisite matrix), signifying a prerequisite relation between KC_i and KC_j with $KC_{ij} = 1$. Joint graph convolutional network based deep KT [45] utilizes the homogeneous weighted graph to delineate the co-occurrence relations of KCs, where the weights between nodes (KCs) quantify their co-occurrence degrees. Conversely, structure based KT [42] employs a heterogeneous graph to articulate both the prerequisite and similarity relations among KCs, with prerequisite relation modeling akin to PDKT-C and similarity relations modeled based on a threshold, focusing on KCs relations where the similarity exceeds 5.0.

Pre-defined graph necessitates substantial domain knowledge and is sensitive to the graph quality. Typically, graphs constructed in this manner are intuitive yet incomplete, and not explicitly tailored to specific prediction tasks. They might incorporate biases and lack adaptability in domains where suitable knowledge is not available. Instead of defining the graph manually, GKT [43] proposes adaptive graph learning to model the relations of KCs in a data-driven manner. Specifically, GKT introduces three adaptive mechanisms to infer the KC relations: 1) Parametric adjacency matrix indicates that the relations in PAM are randomly initialized parameters and are optimized during training. 2) Multi-head attention indicates that the relations in MHA are the average weighted attention scores. 3) Variational autoencoder indicates that the relations in VAE are the latent discrete variables sampled from Gumbel-Softmax distribution [95].

In the modeling of associations among KCs, both pre-defined and adaptive graphs have their own advantages and disadvantages. The pre-defined graph approach requires domain or prior knowledge to guide the construction of the graph and has poor generalizability, necessitating separate consideration for different datasets. However, its advantage lies in computational efficiency. On the other hand, the adaptive graph approach is straightforward to implement and has better generalizability, but it requires a substantial amount of data for training to achieve satisfactory results and demands higher computational resources compared to the predefined graph approach.

3) Question-Side Information: In the auxiliary knowledge base, KCs and questions are closely related yet distinct: questions are variations and combinations built upon KCs, and KCs are the underlying building elements of questions. Typically, in a KT dataset, the number of KCs is significantly fewer than that of questions. Therefore, two different questions can correspond to exactly the same KCs. Solely relying on KC-side information may not effectively measure the differences in students' performance when they answered different questions. Therefore, existing studies on processing question-side information mainly focus on two aspects: one is the extraction of features inherent to the question itself, and the other is the extraction of associative features between questions and KCs.

Question data itself is multimodal, including text descriptions, formulas and images. Therefore, some studies aim to represent questions to extract effective features for downstream KT tasks. For example, EERNN [58] argues that the traditional method of substituting questions with KCs loses crucial information hidden within the questions, such as difficulty level. To address this problem, EERNN embeds the words in the text descriptions of questions into vector embeddings by using word2vec [56]. Likewise, AdaptKT [60] also learns the textual question features with word2vec. EHFKT [62] utilizes BERT [57] to encode the textual information within questions. RKT [87] and option tracing KT model [96] use the linear layer to generate the words in questions into the vector embeddings. QuesNet [64] uses CNN and MLP to embed text information, image information, and metadata information. ERAKT [65] introduces an innovative embedding method for formulas. It uniformly substitutes LaTeX formulas using an ontology replacement technique and conducts consistent preprocessing along with other texts.

Moreover, some researchers have noticed the relations between questions and KCs and attempted to construct graphs to mine high-order relations. GIKT [50] constructs a question-KC graph to exploit high-order graph-level relations to mitigate the data sparsity problem and the multi-KC issue. HGKT [69] designs a hierarchical graph structure to encapsulate direct and indirect support relationships among questions. It comprises two layers: the direct support graph at the bottom and the indirect support graph at the top, each representing a distinct type of question-question relationship. The lower layer directly correlates to questions, while the upper layer corresponds to question schema. Question schema defined in HGKT is a group of similar questions with similar solutions. Bi-graph contrastive learning based KT (Bi-CLKT) [44] constructs a question-level influence graph to model the relations between questions. Unlike HGKT, which employs an unweighted graph to represent relationships, Bi-CLKT utilizes a weighted graph to measure the co-correctness rate between two questions.

E. Learning Outcome Objective

While classification is commonly used as a means to construct loss functions in KT tasks, it is important to emphasize that KT itself is not merely a classification problem. Although classification accuracy is a widely adopted metric for evaluating KT models, the fundamental objective of KT is to model and understand the dynamic cognitive states of students, rather than solely optimizing for classification performance. By focusing on the cognitive processes underlying student learning, KT models can provide deeper insights into the educational journey of students, making them powerful tools not just for prediction, but for fostering personalized learning experiences and informed educational decision-making.

This section first introduces the basic KT loss schemas of the negative log-likelihood loss function and the cross-entropy loss function, commonly employed in machine learning tasks, including KT. However, beyond these foundational loss functions,

KT models play a crucial role in constructing cognitive profiles of students. Cognitive profiles encompass detailed representations of a student's knowledge acquisition over time, identifying areas of strength, challenges, and potential learning trajectories. By analyzing patterns in student responses, KT models can infer the mastery levels of various KCs, predict future learning difficulties, and help educators design targeted interventions.

The Negative Log-Likelihood and Cross-Entropy Loss Functions are two commonly employed loss functions in machine learning tasks, frequently utilized in KT tasks as well. Several studies have endeavoured to devise specialized constraints based on these two loss functions to augment the efficacy of KT tasks. However, the most significant contribution of KT models lies in their ability to model assumptions about student learning, which is crucial for constructing accurate cognitive profiles. For instance, PDKT-C [94] not only predicts the correctness of student responses but also models prerequisite relationships between KCs, providing a more nuanced understanding of a student's cognitive state.

Specifically, PDKT-C posits that current research on prerequisites primarily focused on identifying these prerequisites rather than leveraging them to assist in KT. It proposes a hypothesis: at time t, if student i had mastered KC a, then he or she must also has mastered a's prerequisite, KC b. Similarly, if at time t, student i had not mastered the prerequisite b of KC a, then he or she is likely not to have mastered KC a either. PDKT proposes an ordering pair to formulate the hypothesis:

$$P\left(m_{i,k_2,t_2}=1\right) \leq P\left(m_{i,k_1,t_1}=1\right), \tag{3}$$

where $m_{i,k,t}$ represents the binary value of whether student i masters concept k at time t. The final loss function is composed of two parts: one part is the prediction error regarding whether the student's answer is correct or not, and the other part is the constraint on prerequisites. Specifically, the loss function can be represented as:

$$P\left(m_{i,k_2,t_2}=1\right) \leq P\left(m_{i,k_1,t_1}=1\right), \tag{4}$$

where $\Theta = \{\mathbf{c}_k \mid k \in \mathcal{C}\} \cup \Theta_{GRU}$ denotes the overall set of parameters.

Similarly, Yeung et al. [74] identified two primary issues in the prediction outcomes of DKT and proposed a novel model named DKT+. First, it occasionally fails to reconstruct the input observation due to counterintuitive prediction results. For instance, when a student correctly or incorrectly answer a question related to skill s_i, the predicted probability of the student answering s_i correctly could sometimes paradoxically decrease or increase. Second, the predicted knowledge state fluctuates and lacks consistency over time, which is not ideal as a student's knowledge state that is expected to transition gradually and consistently. To address the consistency issue in DKT's predictions, they designed three regularization terms: reconstruction error r to tackle the reconstruction issue, and waviness measures w_1 and w_2 to smooth the transition of the predicted knowledge state.

DKT+ first suggestes a reconstruction error r to tackle the reconstruction issue:

$$r = \frac{1}{\sum_{i=1}^{n}\left(T_i-1\right)}\left(\sum_{i=1}^{n}\sum_{t=1}^{T_i-1} l\left(\mathbf{y}_t^i \cdot \boldsymbol{\delta}\left(q_t^i\right), a_t^i\right)\right), \tag{5}$$

where n represents the total number of students, T_i is the number of interactions of student i, y_i^t is the predicted result of student i at time t, a_i^t is the actual answer of student i at time t, and q_i^t is the question label of student i at time t. δ is a function that returns a vector, which has 1 at the position corresponding to q_i^t and 0 elsewhere. l is the loss function. Moreover, it proposes waviness measures w_1 and w_2 to smooth the transition of the predicted knowledge state:

$$\begin{aligned} w_1 &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T_i-1}\left\|\mathbf{y}_{t+1}^i-\mathbf{y}_t^i\right\|_1}{M\sum_{i=1}^{n}\left(T_i-1\right)} \\ w_2^2 &= \frac{\sum_{i=1}^{n}\sum_{t=1}^{T_i-1}\left\|\mathbf{y}_{t+1}^i-\mathbf{y}_t^i\right\|_2^2}{M\sum_{i=1}^{n}\left(T_i-1\right)}, \end{aligned} \tag{6}$$

where n represents the total number of students, T_i is the number of interactions of student i, and y_i^t is the predicted result of student i at time t. $\|\cdot\|_1$and $\|\cdot\|_2$ denote the L1 and L2 norms, respectively.

Finally, in the context of modeling student knowledge states, DHKT [66] introduces a grouping mechanism to ensure that a student's responses to questions reflecting the sameKC exhibit similarity. This mechanism implies that questions about the same knowledge should be considered as part of a unified cognitive profile, where each group's questions resemble each other closely in the feature space. To enforce this similarity, it introduces a hinge loss function aimed at maximizing the margin between different groups, thereby enhancing the model's ability to distinguish between various knowledge clusters:

$$l_h^{m,n} = \begin{cases} \max\left\{0, 1-\mathbf{e}_{i_n}^T\mathbf{e}_{c_m}\right\} & q_{m,n}=1 \\ \max\left\{0, 1+\mathbf{e}_{i_n}^T\mathbf{e}_{c_m}\right\} & q_{m,n}=0 \end{cases}, \tag{7}$$

where $\mathbf{e}_{i_n}$ and $\mathbf{e}_{c_m}$ are the embeddings of the question i_n and concept c_m, and $q_{m,n}$ is an indicator in the question to concept mapping matrix $\mathcal{Q}$ indicating whether the question n is related to concept m. The purpose of this loss function is to ensure that questions within the same group remain proximate to each other in the feature space while being distanced from questions in other groups, thereby enhancing the model's capacity to build accurate cognitive profiles.

Similarly, qDKT [33] emphasizes that for a given learner, the success probabilities across multiple questions related to the same skill should not exhibit significant disparities. This constraint implies that if two questions are both predicated on the same skill, then a student's performance on these two questions should be similar. To maintain consistency within cognitive profiles, qDKT proposes a regularization term to constrain the variance of success probabilities for questions under the same skill:

$$R(\boldsymbol{y}) = \sum_{i\in Q}\sum_{j\in Q}\mathbf{1}(i,j)\cdot\left(y_i-y_j\right)^2, \tag{8}$$

where vector $\boldsymbol{y} \in R^N$ contains success probabilities of all questions Q in the dataset, $i, j \in Q$ and $\mathbf{1}(i, j)$ is 1 if i, j fall under the same skill, otherwise it is 0.

These examples demonstrate that while classification tasks are integral to KT model development, the broader aim is to construct and apply cognitive profiles that not only predict student performance but also inform educational decision-making.

F. Computational Efficiency and Scalability

In Section IV-C, we categorize the main architectures of KT models into four types: sequence-based, memory-based, attention-based, and forgetting-based. In this subsection, we conduct a detailed review on the computational efficiency and scalability of these four types of model architectures.

1) Computational Efficiency: When considering the complexity of a KT model, two points must be noted: 1) The overall computational complexity of a model is not solely determined by its primary architecture, though the architecture itself constitutes a significant portion of the computational burden; 2) The complexity of individual models within each category may vary, but this analysis focuses on the computational complexity at the architectural level.

Our analysis of the computational efficiency of KT models is presented as follows, following the categorization previously established in Section IV-C.

- *Sequence-based Student Knowledge Memory Modeling:* For RNN or LSTM-based models, the complexity is typically $O(n \cdot d^2)$, where n is the sequence length and d is the hidden layer size. If the model uses self-attention, the complexity is generally $O(n^2 \cdot d)$ due to the pairwise attention scores computation. Additionally, if the model includes memory units for each knowledge point, the complexity may increase to $O(m \cdot d)$, where m is the number of memory units. Therefore, the overall complexity is a combination of these factors, usually represented as $O(n \cdot d^2) + O(n^2 \cdot d) + O(m \cdot d)$.
- *Memory-based Student Knowledge Memory Modeling:* For memory-based student knowledge memory modeling, the computational complexity primarily arises from updating the memory units associated with each knowledge point. Assuming there are m knowledge points and each has a corresponding memory unit of dimension d, the complexity of updating all memory units is $O(m \cdot d)$. For a sequence with n time steps, the overall computational complexity is approximately $O(n \cdot m \cdot d)$. If the memory updates involve more complex operations, such as using LSTM to update memory units, the complexity could increase to $O(n \cdot m \cdot d^2)$.
- *Attention-based Student Knowledge Memory Modeling:* The computational complexity of attention-based student knowledge memory modeling is primarily driven by the self-attention mechanism, which typically has a complexity of $O(n^2 \cdot d)$, where n is the sequence length and d is the hidden layer dimension. This complexity arises because the model computes the relevance between each time step in the sequence. If the model employs multi-head attention, the overall complexity increases to $O(h \cdot n^2 \cdot d)$, where h is the number of attention heads.
- *Forgetting-based Student Knowledge Memory Modeling:* The computational complexity of forgetting-based student knowledge memory modeling depends on the forgetting mechanism used. If a simple time decay function, such as exponential decay, is employed, the complexity is typically $O(n)$, where n is the sequence length, as each time step requires a single calculation. If the model uses RNNs(e.g., LSTM) to simulate the forgetting process, the complexity for each time step increases to $O(d^2)$, resulting in an overall complexity of $O(n \cdot d^2)$. Thus, the complexity for forgetting-based models can range from $O(n)$ to $O(n \cdot d^2)$, depending on the complexity of the forgetting mechanism.

Overall, attention-based models tend to have the highest computational complexity, especially with long sequences, while forgetting-based models can be the most efficient depending on the forgetting mechanism employed.

2) Scalability: In this subsection, we evaluate the scalability of the four types of student knowledge memory modeling architectures: sequence-based, memory-based, attention-based, and forgetting-based. Below is a detailed analysis of how each architecture performs in terms of scalability:

- *Sequence-based Student Knowledge Memory Modeling:* The scalability of sequence-based models is often limited by the length of the sequence, especially when using RNN or LSTM architectures. As the sequence length increases, the demand for computational resources and time also significantly increases. These models are difficult to parallelize because each time step's computation depends on the output of the previous step. While suitable for shorter sequences, their scalability is constrained when handling longer sequences or large-scale datasets. Models using self-attention mechanisms perform better in parallel computation, and despite their complexity growing quadratically with sequence length, they exhibit better scalability on large-scale data due to efficient parallel processing on hardware such as GPUs.
- *Memory-based Student Knowledge Memory Modeling:* The scalability of memory-based models depends on the number of memory units that the model needs to maintain. As the number of knowledge points (memory units) increases, the storage and computational requirements also grow, potentially leading to bottlenecks when processing large-scale educational datasets, especially if memory units need frequent updates. However, scalability can be improved through distributed computing or memory compression techniques. These models are effective in scenarios with a fixed number of knowledge points but may face limitations when dealing with dynamically increasing knowledge points.
- *Attention-based Student Knowledge Memory Modeling:* Attention-based models excel in scalability and parallel processing, as the attention mechanism allows different time steps to be processed simultaneously across multiple computational cores. This makes these models particularly

TABLE II
DATA STATISTICS OF 9 DATASETS IN PYKT

Datasets	Original					After Preprocessing					
	Interactions	Qequences	Questions	KCs	AvgKCs	Interactions	Sequences	Questions	KCs	AvgKCs	Subject
Statics 2011	194,947	333	-	1224	-	189,297	1,034	-	1223	-	Physics
ASSIST 2009	346,860	4,217	26,688	123	1.1969	337,415	4,661	17,737	123	1.1970	Math
ASSIST 2015	708,631	19,917	-	100	-	682,789	19,292	-	100	-	Math
Algebra 2005	809,694	574	210,710	112	1.3634	884,098	4,712	173,113	112	1.3634	Math
Bridge2006	3,679,199	1,146	207,856	493	1.0136	1,824,310	9,680	129,263	493	1.0136	English
Ednet-small	597,042	4,999	11,901	188	2.2449	597,042	4,999	11,901	188	2.2449	English
Ednet-large	5,936480	50000	12235	188	2.2600	140147634	1057441	12235	188	2.26	Math
NIPS34	1,382,727	4,918	948	57	1.0148	1,399,470	9,401	948	57	1.0148	Math
POJ	996,240	22,916	-	2750	-	987,593	20,114	-	2748	-	Programming
XES3G5M	5,549,635	18,066	7,652	865	1.164	6,413,318	41,676,	7,652	865	1.16	Math

"Original" and "After Preprocessing" refer to initial and preprocessed data statistics. "AvgKCS" denotes the number of average KCs per quesiton. In order to accommodate the large size of the original Ednet dataset, a random selection of 4,999 and 50,000 student sequences was performed, resulting in the creation of two subsets named Ednet-small and Ednet-large respectively.

well-suited for large-scale datasets and long sequences. However, as sequence length increases, the computational complexity grows quadratically, which can pose scalability challenges in extremely large datasets. Due to the high parallelism of the attention mechanism, these models typically scale well on modern hardware such as GPUs or TPUs but require substantial memory to store large-scale attention matrices.

- *Forgetting-based Student Knowledge Memory Modeling:* The scalability of forgetting-based models is closely tied to the forgetting mechanism employed. Models using simple time decay functions offer excellent scalability due to their relatively low computational demands, making them efficient for handling large-scale datasets. However, if more complex mechanisms(e.g., LSTM) are used to simulate the forgetting process, computational complexity and time requirements increase significantly, which may impact scalability. Overall, forgetting-based models scale well when handling fixed-length sequences or predictable forgetting patterns, but their scalability may be limited when facing complex learning behaviors or large-scale dynamic datasets.

Overall, attention-based models generally exhibit the best scalability, particularly when handling large-scale data in parallel computing environments. Sequence-based models perform well with short sequences and small datasets but may encounter bottlenecks when scaling to longer sequences. The scalability of Memory-based models is primarily constrained by the number of memory units and the frequency of updates. In contrast, Forgetting-based models demonstrate good scalability with simple forgetting mechanisms, but complex forgetting mechanisms may limit their applicability to large-scale datasets.

V. A REPOSITORY FOR DEEP LEARNING BASED KNOWLEDGE TRACING

To foster advancements in KT methodologies, we have meticulously developed PYKT, a detailed Python-based benchmark library tailored to meet the demands of real-world KT applications within educational settings [97]. The PYKT library consists of a standardized set of intefrated data preprocessing procedures on 9 popular datasets across different domains, 5 detailed prediction scenarios, and 21 frequently compared DLKT models for transparent and extensive experiments. The PYKT library is released at https://github.com/pykt-team/pykt-toolkit.

A. Datasets

PYKT provides a collection of public datasets suitable for evaluating KT models. These datasets originate from online educational platforms, where copious student learning data is captured to study their learning behaviors and knowledge proficiency. Table II presents a detailed list of these datasets, accompanied by essential information and statistics. Collected across diverse learning scenarios, these datasets exhibit notable variations in both scale and subject content. Utilizing the PYKT Python library, researchers can effortlessly download and preprocess these datasets. Next, we provide a detailed introduction to the data included in PYKT.

ASSISTments Datasets: The ASSISTments datasets comprise longitudinal data sourced from the ASSISTment online tutoring platform. Commonly used as benchmarks for KT models, these datasets encompass a vast array of questions. Several versions of the ASSISTments datasets exist, each corresponding to data amassed during distinct time frames. Details for each version are elucidated as follows:

- *ASSISTments2009:*[3] This dataset is made up of math exercises, collected from the free online tutoring ASSISTments platform in the school year 2009-2010. As discussed in [10], the initial dataset of ASSISTments2009 contains several issues that can significantly compromise the reliability of the experiment. However, these problems have been effectively addressed in the subsequent upgraded version called 'skill-builder'. The latest version of dataset consists of 346,860 interactions, 4,217 students, and 26,688 questions, which is widely used and has been the standard benchmark for KT methods over the last decade.
- *ASSISTments2015:*[4] Analogous to ASSISTments2009, this dataset is amassed from the ASSISTments platform during 2015. It captures 708,631 interactions across

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

[4]https://sites.google.com/site/assistmentsdata/datasets/2015-assistments-skill-builder-data

100 unique KCs involving 19,917 students. Notably, this dataset boasts the highest student count among all other ASSISTments datasets. It is essential to highlight that this collection exclusively contains records of student responses to the 100 KCs, omitting details about the exercises.

STATICS2011:[5] This dataset is collected from an engineering statics course taught at the Carnegie Mellon University during Fall 2011. In this dataset, a unique question is constructed by concatenating the problem name and step name and the dataset has 194,947 interactions, 333 students, and 1,224 questions.

Algebra2005:[6] This dataset is from the KDD Cup 2010 EDM Challenge that contains 13-14 year old students' responses to Algebra questions. It contains detailed step-level student responses. The unique question construction is similar to the process used in Statics2011, which ends up with 809,694 interactions, 574 students, 210,710 questions, and 112 KCs.

Bridge2006:[7] This dataset is also from the KDD Cup 2010 EDM Challenge and the unique question construction is similar to the process used in Statics2011. There are 3,679,199 interactions, 1,146 students, 207,856 questions, and 493 KCs in the dataset.

EdNet:[8] The EdNet dataset is a collection of learning records from students studying English using the AI tutoring system Santa in South Korea. It contains data from over two years of students preparing for the TOEIC Listening and Reading Test. EdNet is currently the largest publicly available dataset in the field of KT, with a total of 131,441,538 learning records from 784,309 students.

NIPS34:[9] This dataset is from the tasks 3 and 4 at the NeurIPS 2020 Education Challenge. It contains students' answers to multiple-choice diagnostic math questions and is collected from the Eedi platform. For each question, we choose to use the leaf nodes from the subject tree as its KCs, which ends up with 1,382,727 interactions, 948 questions, and 57 KCs.

POJ:[10] This dataset consists of programming exercises and is collected from Peking coding practice online platform. The dataset is originally scraped by Pandey and Srivastava. In total, it has 996,240 interactions, 22,916 students, and 2,750 questions.

XES3G5M:[11] A large-scale dataset is collected from a real-world online math learning platform, which contains 7,632 questions, 865 KCS, and 5,549,635 interactions from 18,066 students.

B. Evaluation Procedure

A standardized evaluation protocol is a fundamental component of AI research, exerting significant influence on model performance, fairness, and robustness. Inconsistencies in the training and evaluation procedures, even when employing the same public KT dataset, can yield unexpectedly disparate outcomes. For example, the AUC scores of DKT and AKT on ASSISTments2009 range from 0.73 to 0.821 and from 0.747 to 0.835 in respectively. To end this, PYKT provides a standardized assessment process, it is closer to the researchers expect KT application in the education environment.

1) Data Preprocessing: The KT datasets mentioned previously require extensive preprocessing steps to make them usable, including the removal of duplicates, null values, and incorrect entries, as well as the reconstruction of the time series. Hence, PYKT [97] benchmark outline a reasonable data preprocessing procedure for DLKT research, which is listed as follows:

- Data Filter: filter out interactions from the dataset if they do not have a student ID or any type of information in 4-tuple interaction representation[12] is not available or missing. Additionally, filter out students from the dataset if their sequences have less than 3 interactions.
- Data Splitting: 80% of the dataset is randomly selected and divided into five folds, with four folds as the training set and one fold as the validation set. The remaining 20% is used as the test set.
- KC Subsequence Generation: For training and validation, the original question-response sequence is expanded into KC-level by repeating responses for questions with multiple KCs. The expanded KC-level response sequence is then truncated into subsequences of length m, where m represents the maximum training sequence length. Subsequences shorter than m are padded with -1.

2) Prediction Scenarios: In real-world educational scenarios, the size of the question bank is typically significantly larger compared to the set of KCs used. For instance, in the case of Algebra2005, the number of questions is more than 1500 times greater than the number of KCs (described in Table II). Hence, in order to evaluate the performance of DLKT models based on extremely sparse question-response data, a recommended process is listed as follows:

- *Step 1:* Train DLKT models on KC response data, when a question is associated with a set of KCs, expand each question-level interaction into multiple KC-level interactions to artificially generate data.
- *Step 2:* Use the learned DLKT models to predict on the above expanded KC-response data first, and then the final question-level predictions are output by integrating the predicted mastery levels of their associated KCs.

Although forecasts at both the question and KC levels are important for the development of personalized educational applications, it is recommended to evaluate DLKT models using prediction tasks at the question level rather than the KC level when comparing models offline. This recommendation is based on two reasons. First, student responses are only observable at the question level, which means there is no verifiable ground truth about KCs. Second, a single question may be linked to multiple KCs, which can potentially lead to an overestimation or

[5]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=50700
[6]https://pslcdatashop.web.cmu.edu/KDDCup/
[7]https://pslcdatashop.web.cmu.edu/KDDCup/
[8]https://github.com/riiid/ednet
[9]https://eedi.com/projects/neurips-education-challenge
[10]https://drive.google.com/drive/folders/1LRljqWfODwTYRMPw6wEJ_mMt1KZ4xBDk
[11]https://github.com/ai4ed/XES3G5M

[12]4-tuple interaction representation = <q, {c}, r, t>, where q, {c}, r, t represent the specific question, the related KC set, student's response and student's responds timestamp respectively.

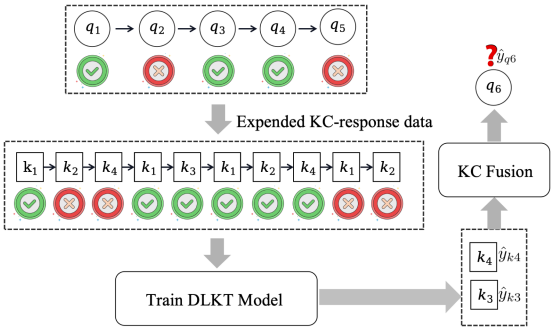

Fig. 7. A recommended procedure for training and evaluating the DLKT models.

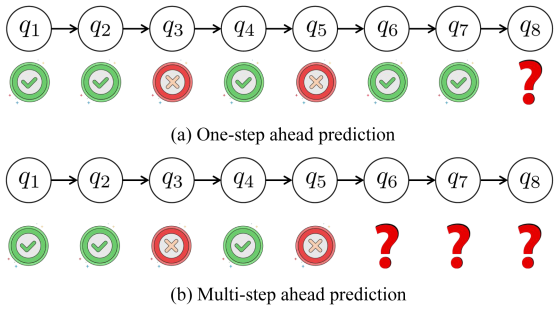

Fig. 8. Two different prediction scenarios: one-step ahead and multi-step ahead predictions.

underestimation of the model's actual performance when evaluated at the KC level. As a result, the PYKT team has come up with a paradigm called 'All-in-One', i.e., simultaneously estimate the mastery level of all the KCs under each specific question. As shown in Fig. 7, to predict the output of q_6 that contains k_3 and k_4, it is necessary to independently predict both $\hat{y}_{k_3}$ and $\hat{y}_{k_4}$. Some existing studies ignore this key issue. The evaluation method used in their open-source project is a One-by-One evaluation method on the expanded KC sequences,i.e., predicting $\hat{y}_{k_{t+1}}$ given all the responses (or labels) of $\hat{y}_{k_1}, \hat{y}_{k_2}, \ldots, \hat{y}_{k_t}$ are known. However, this can cause a label leakage problem. Successive KCs, k_t and k_{t+1}, may belong to the same problem, which could improve the model's performance.

We propose an approach referred to as the 'Train on KC, Test on Question' paradigm for KT training and prediction. This paradigm can partially alleviate the performance issues in KT prediction tasks caused by data sparsity. However, it does not fully address the underlying data sparsity problem in KT. In recent years, several studies have attempted to tackle data sparsity in KT through contrastive learning [44], [98], [99], [100], as discussed in Section VIII.

3) KC Prediction Aggregation: As mentioned in [97], besides individual KC prediction scenarios, PYKT categorizes related prediction scenarios into the following four types based on the KC Fusion approach: 1) early fusion (EF): this method utilizes the averaged hidden states of all related KCs to predict the question-level response. 2) late fusion-average (LF-AVG): this approach uses the averaged prediction probabilities of all KCs to determine the question-level prediction probability. 3) late fusion-majority vote (LF-MV): this method involves conducting majority votes based on the prediction results of all KCs. 4) late fusion-strict (LF-S): this approach predicts a positive outcome only if all the predicted labels of the associated KCs are positive.

4) One-Step and Multi-Step Ahead KT Predictions: To ensure alignment with practical application scenarios, prediction scenarios are divided into two configurations: 1) one-step forward prediction and 2) multi-step forward prediction. The one-step forward prediction task is designed to forecast the student's response to the final question based on their historical interaction sequence, as shown in Fig. 8(a). Conversely, the multi-step forward prediction task aims to predict a range of student responses based on the student's historical interaction sequence, as shown in Fig. 8(b). Accurate one-step forward prediction can significantly enhance real-time educational recommender systems. Multi-step forward prediction can offer valuable insights for learning path selection and construction, and assist educators in adaptively modifying future teaching materials.

5) Evaluation Metric: The main metric used to evaluate the performance of DLKT models on predicting binary-valued future learner responses to either questions or KCs is AUC. Additionally, accuracy is also considered as a metric for evaluation.

C. Representative DLKT Methods

PYKT encompasses a suite of DLKT algorithms, as delineated below:

- *Deep Sequential KT Models:* DKT [8], DKT+ [74], DKT-F [53], KQN [11], DIMKT [32], qDKT [33], LPKT [38], IEKT [55], QIKT [81], AT-DKT [101]
- *Memory Augmented KT Models:* DKVMN [10], SKVMN [83], DeepIRT [34]
- *Attention based KT Models:* SAKT [7], AKT [6], simpleKT [35], sparseKT [36].
- *Other KT Models:* HawkesKT [91], GKT [43], ATKT [47]

As these algorithms have been delineated in preceding sections, they will not be reiterated.

VI. Empirical Studies

To fairly and effectively compare the performance of current DLKT-related algorithms, we conduct an empirical study based on the PYKT library.

A. Experimental Setup

We addressed the data sparsity issue following the steps outlined in Section V-B2, and according to the procedures in Section V-B1, we preprocessed the data. For datasets ASSISTment2009, Algebra2005, Bridge2006, NIPS34, EdNet and XES3G5M which have both question related and KC related information. For other datasets such as Stastics2011, ASSISTment2015, and POJ which question related or KC related information is missing, both the training and testing procedures are conducted on question-response data Moreover, referring to [97], we choose to utilize LF-AVG for KC prediction fusion, and all experiments are conducted in the one-step ahead

TABLE III
THE OVERALL PERFORMANCE ON AUC

Model	AUC									
	Question-Level (All-in-One)									
	ASSIST2009	Algebra2005	Bridge2006	NIPS34	XES3G5M	EdNet-small	EdNet-large	Statics2011	ASSIST2015	POJ
DKT	0.7541±0.0011	0.8149±0.0011	0.8015±0.0008	0.7689±0.0002	0.7852±0.0006	0.6133±0.0006	0.6420±0.0028	0.8222±0.0013	0.7271±0.0005	0.6089±0.0009
DKT+	0.7547±0.0017	0.8156±0.0011	0.8020±0.0004	0.7696±0.0002	0.7861±0.0002	0.6189±0.0012	0.6429±0.0019	0.8279±0.0004	0.7285±0.0006	0.6173±0.0007
DKT-F	-	0.8147±0.0013	0.7985±0.0013	0.7733±0.0003	0.7940±0.0006	0.6168±0.0019	0.6548±0.0035	0.7839±0.0061	-	0.6030±0.0023
KQN	0.7477±0.0011	0.8027±0.0015	0.7936±0.0014	0.7684±0.0003	0.7793±0.0006	0.6111±0.0022	0.6415±0.0014	0.8232±0.0007	0.7254±0.0004	0.6080±0.0015
DKVMN	0.7473±0.0006	0.8054±0.0011	0.7983±0.0009	0.7673±0.0004	0.7792±0.0004	0.6158±0.0022	0.6406±0.0044	0.8093±0.0017	0.7227±0.0004	0.6056±0.0022
ATKT	0.7470±0.0008	0.7995±0.0023	0.7889±0.0008	0.7665±0.0001	0.7783±0.0004	0.6065±0.0003	0.6495±0.0013	0.8055±0.0020	0.7245±0.0007	0.6075±0.0012
GKT	0.7424±0.0021	0.8110±0.0009	0.8046±0.0008	0.7689±0.0024	0.7727±0.0006	0.6223±0.0017	0.6372±0.0029	0.8040±0.0065	0.7258±0.0012	0.6070±0.0036
SAKT	0.7246±0.0017	0.7880±0.0063	0.7740±0.0008	0.7517±0.0005	0.7693±0.0008	0.6072±0.0018	0.6262±0.0017	0.7965±0.0014	0.7114±0.0003	0.6095±0.0013
SAINT	0.6958±0.0023	0.7775±0.0017	0.7781±0.0013	0.7873±0.0007	0.8074±0.0007	0.6614±0.0019	0.6818±0.0041	0.7599±0.0139	0.7026±0.0011	0.5563±0.0012
AKT	0.7853±0.0017	0.8306±0.0019	0.8208±0.0007	0.8033±0.0003	0.8207±0.0008	0.6721±0.0022	0.6911±0.0075	0.8309±0.0009	0.7281±0.0004	0.6281±0.0013
SKVMN	0.7332±0.0009	0.7463±0.0022	0.7287±0.0052	0.7513±0.0005	0.7514±0.0005	0.6182±0.0114	0.6412 ±0.0092	0.8086±0.0024	0.7080±0.0004	0.6000±0.0024
HAWKES	0.7224±0.0006	0.8210±0.0012	0.8068±0.0010	0.7767±0.0010	0.7921±0.0007	0.6837±0.0016	0.7307±0.0009	-	-	-
DeepIRT	0.7465±0.0006	0.8040±0.0013	0.7976±0.0006	0.7672±0.0006	0.7785±0.0005	0.6173±0.0008	0.6401±0.0026	0.8051±0.0041	0.7219±0.0003	0.6042±0.0020
LPKT	0.7812±0.0023	0.8268±0.0004	0.8056±0.0008	0.8004±0.0003	0.8163±0.0002	0.7392±0.0011	0.7644±0.0006	-	-	-
DIMKT	0.7717±0.0011	0.8277±0.0009	0.8167±0.0008	0.8030±0.0002	0.8220±0.0002	0.6748±0.0030	0.7070±0.0042	-	-	-
IEKT	0.7861±0.0027	0.8416±0.0014	0.8125±0.0009	0.8045±0.0002	0.8280±0.0002	0.7335±0.0014	0.7649±0.0004	-	-	-
qDKT	0.7016±0.0049	0.7485±0.0017	0.7524±0.0005	0.7995±0.0008	0.8225±0.0002	0.6987±0.0010	0.7497±0.0004	-	-	-
AT-DKT	0.7555±0.0005	0.8246±0.0019	0.8104±0.0009	0.7816±0.0002	0.7932±0.0004	0.6249±0.0020	0.6585±0.0040	-	-	-
simpleKT	0.7744±0.0018	0.8254±0.0003	0.8160±0.0006	0.8035±0.0000	0.8163±0.0006	0.6599±0.0027	0.6807±0.0054	0.8199±0.0011	0.7248±0.0005	0.6252±0.0005
QIKT	0.7878±0.0024	0.8408±0.0007	0.8101±0.0003	0.8044±0.0005	0.8222±0.0006	0.7271±0.0012	0.7594±0.0001	-	-	-
sparseKT-soft	0.7739±0.0005	0.8152±0.0030	0.8120±0.0008	0.8033±0.0008	0.8165±0.0015	0.6628±0.0036	0.6819±0.0035	0.8166±0.0013	0.7379±0.0019	0.6323±0.0034
sparseKT-topK	0.7681±0.0024	0.8100±0.0038	0.8117±0.0009	0.8043±0.0004	0.8198±0.0004	0.6626±0.0033	0.6819±0.0017	0.8197±0.0011	0.7502±0.0004	0.6401±0.0013

TABLE IV
THE OVERALL PERFORMANCE ON ACCURACY

Model	Accuracy									
	Question-Level (All-in-One)									
	ASSIST2009	Algebra2005	Bridge2006	NIPS34	XES3G5M	EdNet-small	EdNet-large	Statics2011	ASSIST2015	POJ
DKT	0.7244±0.0014	0.8097±0.0005	0.8553±0.0002	0.7032±0.0004	0.8173±0.0002	0.6462±0.0028	0.6603±0.0033	0.7969±0.0006	0.7503±0.0003	0.6328±0.0020
DKT+	0.7248±0.0009	0.8097±0.0007	0.8553±0.0003	0.7039±0.0004	0.8178±0.0001	0.6571±0.0019	0.6664±0.0011	0.7977±0.0006	0.7510±0.0004	0.6482±0.0021
DKT-F	-	0.8090±0.0005	0.8536±0.0004	0.7076±0.0002	0.8209±0.0003	0.6402±0.0021	0.6666±0.0031	0.7872±0.0011	-	0.6371±0.0030
KQN	0.7228±0.0009	0.8025±0.0006	0.8532±0.0006	0.7028±0.0001	0.8152±0.0002	0.6422±0.0043	0.6656±0.0007	0.7978±0.0007	0.7500±0.0003	0.6435±0.0017
DKVMN	0.7199±0.0010	0.8027±0.0007	0.8545±0.0002	0.7016±0.0005	0.8155±0.0001	0.6444±0.0030	0.6581±0.0036	0.7929±0.0006	0.7508±0.0006	0.6393±0.0015
ATKT	0.7208±0.0009	0.7998±0.0019	0.8511±0.0004	0.7013±0.0002	0.8145±0.0002	0.6369±0.0009	0.6642±0.0021	0.7904±0.0011	0.7494±0.0002	0.6332±0.0023
GKT	0.7153±0.0032	0.8088±0.0008	0.8555±0.0002	0.7014±0.0028	0.8135±0.0004	0.6625±0.0064	0.6649±0.0021	0.7902±0.0021	0.7504±0.0010	0.6117±0.0147
SAKT	0.7063±0.0018	0.7954±0.0020	0.8461±0.0005	0.6879±0.0004	0.8124±0.0002	0.6391±0.0041	0.6474±0.0044	0.7879±0.0015	0.7474±0.0002	0.6407±0.0035
SAINT	0.6936±0.0034	0.7791±0.0016	0.8411±0.0065	0.7180±0.0006	0.8177±0.0006	0.6522±0.0024	0.6607±0.0028	0.7682±0.0056	0.7438±0.0010	0.6476±0.0003
AKT	0.7392±0.0021	0.8124±0.0011	0.8587±0.0005	0.7323±0.0005	0.8273±0.0007	0.6655±0.0042	0.6738±0.0075	0.8021±0.0011	0.7521±0.0005	0.6492±0.0010
SKVMN	0.7156±0.0012	0.7837±0.0023	0.8406±0.0005	0.6885±0.0005	0.8075±0.0003	0.6555±0.0152	0.6798±0.0072	0.7924±0.0011	0.7459±0.0005	0.6412±0.0021
HAWKES	0.7046±0.0008	0.8115±0.0009	0.8559±0.0005	0.7110±0.0007	0.8188±0.0003	0.6917±0.0013	0.7075±0.0006	-	-	-
DeepIRT	0.7195±0.0004	0.8037±0.0009	0.8543±0.0003	0.7014±0.0008	0.8150±0.0002	0.6457±0.0033	0.6571±0.0031	0.7908±0.0024	0.7509±0.0003	0.6373±0.0006
LPKT	0.7355±0.0013	0.8154±0.0008	0.8547±0.0005	0.7309±0.0006	0.8264±0.0001	0.7146±0.0014	0.7243±0.0004	-	-	-
DIMKT	0.7354±0.0019	0.8109±0.0005	0.8579±0.0001	0.7312±0.0005	0.8291±0.0006	0.6699±0.0038	0.6812±0.0029	-	-	-
IEKT	0.7375±0.0042	0.8236±0.0010	0.8553±0.0023	0.7330±0.0002	0.8316±0.0002	0.7123±0.0007	0.7239±0.0002	-	-	-
qDKT	0.6787±0.0039	0.7262±0.0012	0.8301±0.0007	0.7299±0.0007	0.8301±0.0000	0.6922±0.0004	0.7142±0.0004	-	-	-
AT-DKT	0.7250±0.0007	0.8144±0.0008	0.8560±0.0005	0.7146±0.0002	0.8198±0.0004	0.6512±0.0039	0.6718±0.0034	-	-	-
simpleKT	0.7320±0.0012	0.8083±0.0005	0.8579±0.0003	0.7328±0.0001	0.8246±0.0005	0.6557±0.0029	0.6651±0.0032	0.7957±0.0020	0.7508±0.0004	0.6522±0.0008
QIKT	0.7381±0.0014	0.8222±0.0006	0.8539±0.0005	0.7333±0.0005	0.8300±0.0005	0.7082±0.0016	0.7207±0.0004	-	-	-
sparseKT-soft	0.7282±0.0016	0.8017±0.0020	0.8569±0.0006	0.7322±0.0012	0.8234±0.0009	0.6556±0.0048	0.6626±0.0018	0.7953±0.0013	0.7547±0.0011	0.6549±0.0019
sparseKT-topK	0.7274±0.0022	0.7997±0.0016	0.8571±0.0007	0.7325±0.0013	0.8255±0.0003	0.6576±0.0028	0.6631±0.0015	0.7969±0.0010	0.7597±0.0009	0.6565±0.0024

prediction setting. LF-AVG is an abbreviation for late fusion - average, which means it uses the averaged prediction probabilities of all KCs as the question-level prediction probability. All experimental setups followed those suggested in PYKT [97].

B. Overall Performance

The comprehensive performance of all commonly used DLKT models on several widely adopted KT datasets is presented in Tables III and IV. The following is a summary of the detailed results (when features required by the model are not included in some data sets, the results are replaced with -).

1) Experimental Performance: From Table III, we can find that: 1) Based on the AUC values, the best model is IEKT, with an average AUC of 0.79285. This model outperforms (almost) the other models in terms of AUC. 2) Deep sequential models, i.e., DKT, DKT+, DKT-F and KQN outperform the self-attention based model SAKT and SAINT in majority cases. This is different from most NLP tasks. This indicates that faraway history interactions have limited predictive value in determining students' future performance. 3) DKT, as the first DLKT model, still maintains a remarkable performance among all the proposed models so far.

Our insights based on the experimental results are as follows:

- *Sequence-based and Attention-based Models as Optimal Choices:* The experimental results, as illustrated in Table III, indicate that methods related to memory-based student knowledge memory modeling, such as DKVMN and SKVMN, do not perform prominently. When considering overall performance across various datasets, sequence-based and attention-based student knowledge memory modeling models emerge as the most effective.

TABLE V
THE BOOSTED DLKT AUC RESULTS DUE TO LABEL LEAKAGE

Method	KC Level (One-by-One)				KC Level (All-in-One)				Exaggerated Performance Gains			
	AS09	NIPS	AL05	BD06	AS09	NIPS	AL05	BD06	AS09	NIPS	AL05	BD06
DKT	0.8262	0.7742	0.9218	0.8028	0.7419	0.7681	0.8146	0.8013	0.0843	0.0061	0.1072	0.0015
DKT+	0.8268	0.7748	0.9221	0.8032	0.7424	0.7689	0.8144	0.8019	0.0844	0.0059	0.1077	0.0013
DKT-F	-	0.7787	0.9220	0.7997	-	0.7727	0.8163	0.7984	-	0.0060	0.1057	0.0013
KQN	0.8216	0.7736	0.9179	0.7949	0.7361	0.7677	0.8005	0.7935	0.0855	0.0059	0.1074	0.0014
DKVMN	0.8213	0.7723	0.9190	0.7993	0.7330	0.7668	0.7891	0.7981	0.0883	0.0055	0.1299	0.0012
ATKT	0.8210	0.7718	0.9156	0.7902	0.7337	0.7658	0.7964	0.7885	0.0873	0.0060	0.1192	0.0017
GKT	0.8171	0.7741	0.9208	0.8057	0.7227	0.7681	0.8025	0.8045	0.0944	0.0060	0.1183	0.0014
SAKT	0.7806	0.7532	0.9115	0.7740	0.7085	0.7516	0.7682	0.7738	0.0721	0.0016	0.1433	0.0002
SAINT	0.7605	0.7910	0.9050	0.7787	0.6865	0.7860	0.6662	0.7779	0.0740	0.0050	0.2388	0.0008
AKT	0.8493	0.8084	0.9305	0.8218	0.7650	0.8017	0.8091	0.8206	0.0843	0.0067	0.1214	0.0012

The exaggerated gains are computed by subtracting AUC scores of one-by-one predictions from AUC scores of all-in-one predictions at KC level.

- *Modeling at the Question Level Outperforms Modeling at the KC Level:* The experiments reveal that methods focusing on question-level modeling, such as IEKT, LPKT, and QIKT, outperform those that focus on KC-level modeling. However, due to the vast number of questions in the datasets, these methods incur higher training and inference costs compared to KC-based modeling approaches. In educational scenarios where high accuracy is paramount, question-level modeling methods may be preferable. Conversely, when the differences between questions and KCs are minimal in the dataset, KC-level modeling may be a more efficient choice.
- *The Importance of Forgetting Mechanisms in KT:* Our observations highlight that forgetting behavior is one of the most critical features in current KT. The decay attention mechanism specific to AKT, which accounts for forgetting behavior, is a significant factor in its superior performance among attention-based student knowledge memory modeling methods. We also believe that this focus on short-term dependencies is why RNN-based KT models continue to achieve excellent results, as RNNs inherently emphasize short-term dependencies, and the need for long-term dependency modeling in KT is not as strong as in other domains like time series prediction.
- *Trade-offs Between Efficiency and Performance:* As discussed above, there is currently no KT model that perfectly balances efficiency and performance. Models with superior performance, such as IEKT, AKT, LPKT, and QIKT, are characterized by slower training speeds and higher memory usage. On the other hand, models such as DKT, which train more quickly, do not perform as well as models such as IEKT. In essence, current KT research often sacrifices efficiency for improved model performance. This trade-off necessitates careful selection and balancing of KT models based on specific application scenarios.

2) Impact of Different Subject Areas: From Table III, we can find the following result. 1) KT prediction on programming exercises is much harder compared to KT tasks on math questions. This conclusion can be inferred from the observation that the DLKT models were able to achieve an AUC score of 0.8ish, while they only achieved a score of 0.6ish on the POJ dataset. 2) For English subjects, deep sequential KT models demonstrate superior performance compared to attention-based KT models. This conclusion is substantiated by observations from the EdNet dataset, where the AUC of AKT and other attention-based models falls below 0.7. In contrast, the AUC for IEKT and other sequential models exceeds 0.7. We argue that in EdNet, the average knowledge of each question is greater, which results in the model's attention being distributed more widely and thus affects the model effectiveness.

3) Impact of One-By-One Evaluation Paradigm: As discussed in Section V-B2, when using the One-by-One paradigm for evaluation, label leakage may occur, leading to an inflation of the model's performance. The results of this inaccurate prediction are presented in Table V. As we can see that: 1) Label leakage is particularly severe in the AS09 and AL05 datasets, with an average of 8.38% and 13.09% inflated performance, respectively. This supports our assertion that the One-by-One paradigm results in flat model performance. 2) The diminished influence of label leakage in the NIPS and BD06 datasets can be attributed to their lower Average KC values in comparison to AL05 and AS09, which are quantified as 1.0136 and 1.0148, respectively.

4) Impact of Different KC Fusion Mechanisms: Fig. 7 shows the KC-fusion Module, which includes four fusion ways. We conducted several experiments to evaluate their impact on the final prediction performance empirical (shown in Table VI). DKT, ATKT, and GKT do not use individual hidden states to model each KC. Therefore, the EF approach is not applicable to these methods. Markers $*$, $\circ$ and $\bullet$ are used to indicate whether the LF-AVG model is significantly better, equal to, or worse than the compared method at a 0.01 significance level, respectively. We can find that the performance difference between different fusion mechanisms is very small. The LF-AVG approach outperforms or performs similarly to other approaches across all four datasets.

VII. KNOWLEDGE TRACING BASED EDUCATION APPLICATIONS AND IMPLICATIONS

The application of KT models in educational settings has shown significant potential in transforming how we approach

TABLE VI
IMPACT ON DIFFERENT KC PREDICTION FUSION MECHANISMS

Model	Dataset	Fusion Mechanisms			
		LF-AVG	LF-MV	LF-S	EF
DKT	AS2009	0.7541±0.0011	0.7526±0.0010*	0.7524±0.0012*	-
	AL2005	0.8149±0.0011	0.8123±0.0010○	0.8131±0.0012*	-
	BD2006	0.8015±0.0008	0.8015±0.0008○	0.8015±0.0008○	-
	NIPS34	0.7689±0.0002	0.7687±0.0002*	0.7688±0.0002*	-
DKT+	AS2009	0.7547±0.0017	0.7533±0.0013*	0.7530±0.0022*	-
	AL2005	0.8156±0.0011	0.8124±0.0004*	0.8146±0.0016*	-
	BD2006	0.8020±0.0004	0.8020±0.0004○	0.8020±0.0004○	-
	NIPS34	0.7696±0.0002	0.7695±0.0002*	0.7696±0.0002○	-
DKT-F	AS2009	-	-	-	-
	AL2005	0.8147±0.0013	0.8122±0.0009*	0.8131±0.0016*	-
	BD2006	0.7985±0.0013	0.7985±0.0013○	0.7985±0.0013○	-
	NIPS34	0.7733±0.0003	0.7985±0.0013○	0.7733±0.0003○	-
KQN	AS2009	0.7477±0.0011	0.7457±0.0013*	0.7474±0.0012*	0.7470±0.0011○
	AL2005	0.8027±0.0015	0.7985±0.0016*	0.8012±0.0015*	0.7935±0.0022○
	BD2006	0.7936±0.0014	0.7936±0.0014○	0.7936±0.0014○	0.7936±0.0014○
	NIPS34	0.7684±0.0003	0.7682±0.0003*	0.7684±0.0003○	0.7684±0.0003○
DKVMN	AS2009	0.7473±0.0006	0.7458±0.0006*	0.7456±0.0008*	0.7454±0.0010*
	AL2005	0.8054±0.0011	0.8022±0.0016*	0.8021±0.0009*	0.7961±0.0020*
	BD2006	0.7983±0.0009	0.7983±0.0009○	0.7983±0.0009○	0.7983±0.0010○
	NIPS34	0.7673±0.0004	0.7672±0.0004*	0.7673±0.0004○	0.7673±0.0004○
ATKT	AS2009	0.7470±0.0008	0.7440±0.0007*	0.7466±0.0011*	-
	AL2005	0.7995±0.0023	0.7963±0.0021*	0.7974±0.0026*	-
	BD2006	0.7889±0.0008	0.7888±0.0008*	0.7889±0.0008○	-
	NIPS34	0.7665±0.0001	0.7663±0.0001*	0.7665±0.0001○	-
GKT	AS2009	0.7424±0.0021	0.7376±0.0029*	0.7401±0.002*	-
	AL2005	0.8110±0.0009	0.8072±0.0008*	0.8072±0.0012*	-
	BD2006	0.8046±0.0008	0.8046±0.0008○	0.8046±0.0008○	-
	NIPS34	0.7689±0.0024	0.7686±0.0025*	0.7689±0.0024○	-
SAKT	AS2009	0.7246±0.0017	0.7225±0.0020*	0.7203±0.0016*	0.7193±0.0021○
	AL2005	0.7880±0.0063	0.7801±0.0065*	0.7859±0.0056*	0.7697±0.0097*
	BD2006	0.7740±0.0008	0.7739±0.0008*	0.7740±0.0008○	0.7740±0.0008○
	NIPS34	0.7517±0.0005	0.7516±0.0005*	0.7518±0.0005•	0.7517±0.0005○
SAINT	AS2009	0.6958±0.0023	0.6957±0.0023*	0.6957±0.0023*	0.6957±0.0023○
	AL2005	0.7775±0.0017	0.7041±0.0133*	0.7804±0.0037•	0.6885±0.0145*
	BD2006	0.7781±0.0013	0.7781±0.0013○	0.7781±0.0013○	0.7781±0.0013○
	NIPS34	0.7873±0.0007	0.7870±0.0008*	0.7871±0.0008*	0.7870±0.0009○
AKT	AS2009	0.7853±0.0017	0.7794±0.0009*	0.7847±0.0021*	0.7825±0.0026*
	AL2005	0.8306±0.0019	0.8228±0.0022*	0.8275±0.0019*	0.8177±0.0026*
	BD2006	0.8208±0.0007	0.8208±0.0007○	0.8208±0.0007○	0.8208±0.0007○
	NIPS34	0.8033±0.0003	0.8028±0.0005*	0.8033±0.0003○	0.8034±0.0003•
#win/#tie/#loss		-	31/8/0	20/17/2	6/13/1

learning and teaching. This section explores the broad implications of KT from multiple perspectives, including its impact on students, teachers, and policymakers. These implications are closely related to the components of the proposed GenKT framework. For example, the use of KT models to enhance personalized learning aligns with the 'Student Knowledge Memory' component of GenKT, which focuses on accurately modeling and predicting students' evolving knowledge states. Additionally, the 'Learning Outcome Objective' component of GenKT supports data-driven decision-making for teachers and policymakers, enabling more effective instructional strategies and resource allocation. We delve into how KT can enhance personalized and adaptive learning experiences for students, provide valuable insights and tools for teachers to improve their instructional strategies, and inform data-driven decisions for educational policies. Additionally, we discuss the practical considerations and limitations of applying a unified benchmark for evaluating KT models across different educational contexts, addressing the challenges that arise in diverse and sometimes problematic datasets. These discussions aim to provide a comprehensive understanding of the role of KT in modern education and the considerations necessary for its effective implementation.

A. *Implications for Stakeholders in Education*

KT has aroused increasing interest in research and practice, and it has been employed in various educational scenarios. This subsection demonstrates the potential applications of using KT in education from the perspectives of different stakeholders: students, teachers, and policy-makers.

First, student learning can benefit tremendously from the KT models applied to learning activities. KT models can be used to identify students' strengths, weaknesses, and learning processes, and thus can support personalized learning . Specifically, KT models are appropriate for adaptive learning which requires dynamically measuring and tracking students' learning processes and knowledge states to design personalized learning schemes or instructions based on individual students' abilities [102]. For example, Mohamed Bin Zayed University of Artificial Intelligence utilized KT technology in a graduate-level course, successfully enhancing their ITS system. This integration significantly improved the effectiveness of the video-based learning platform, as evidenced by overwhelmingly positive student feedback [103]. Moreover, the cognitive tutoring system created by researcher at Carnegie Mellon University utilizes KT to display a 'skillometer' a visual representation of a learner's proficiency in various skills pertinent to algebra problem-solving. When a learner seeks a hint or an error is detected, the system promptly updates the KT data and the skillometer [104]. In addition to formal online learning, KT can be applied in the context of game-based learning [105] or assessment [106] to adapt the educational games to the player's current knowledge state and provide personalized feedback [107].

Second, KT can help teachers achieve more effective teaching in instruction, intervention, and assessment and evaluation. Teachers can use KT data to monitor students' learning progress to identify common learning gaps in the class and specific problems in individual students, which can inform classroom instructions and help teachers tailor their teaching strategies to better satisfy the specific needs of each individual student [108]. Teachers are also encouraged to share the KT data with their colleagues to facilitate collaborative teaching and improve the overall instructions. Additionally, by identifying students' weaknesses at the beginning and predicting students' future performance using BKT models [109], teachers can design more targeted and specific intervention programs to support the students to do better in the subsequent tests [110]. Furthermore, KT provides insights into students' knowledge states of specific knowledge contents, allowing teachers to assess students' learning outcomes and evaluate the effectiveness of their classroom instructions.

Third, the implications and applications of KT on teaching and learning also shed light on policy-making. KT can inform policy-makers about the effectiveness of current instructional materials and curricula. Specifically, by analyzing the test data

using KT models, policy-makers are allowed to identify the areas that need improvement and make data-driven decisions about curriculum design [111], [112]. In addition, KT can help policy-makers allocate educational resources, such as funding, facilities, and support services, based on students' needs and learning process, which ensures that the educational resources are allocated to the right places to maximize students' learning outcomes [113].

It is important to note that although the current KT models have been applied in various educational settings and hold significant value for students, teachers, and policymakers, the inherent black-box nature of DLKT models presents a major limitation. Existing DLKT models are unable to provide insight into their decision-making processes, which severely restricts their application. Users not only expect accurate tracking of students' knowledge states but also seek an understanding of the rationale and processes behind the models' decisions. As a result, the interpretability and transparency of DLKT models have become increasingly critical, and there is a growing body of studies focusing on these aspects. For example, Pandey and Karypis introduced SAKT [7], which leverages self-attention to assign relevance weights to past student interactions, thereby enhancing the interpretability of the model's predictions. Lu et al. [114] proposed a method using LRP to enhance the interpretability of RNN-based DLKT models, aiming to make their decision-making processes more transparent. Ding and Larson [115] critically examined the limitations of DLKT models such as DKT and DKVMN in tracking student knowledge, suggesting improvements through attention mechanisms. Zhang et al. [116] introduced the RCKT framework, which utilizes response influence-based counterfactual reasoning to assess the impact of historical responses on future predictions, enhancing model interpretability. Finally, Chen et al. [117] presented the QIKT model, which integrates question-centric cognitive representations with an IRT layer, achieving both superior prediction performance and psychologically meaningful interpretability.

B. Applicability and Limitations of the Unified Benchmark

The unified benchmark framework has demonstrated significant utility in providing consistent and reproducible evaluations of KT models across standardized educational settings. It enables fair comparisons and facilitates the identification of strengths and weaknesses among different DLKT algorithms. However, in specific educational contexts, particularly those involving unique or problematic datasets, the applicability of a unified benchmark may be limited. For instance, datasets with inherent inconsistencies or those that do not align well with the typical assumptions of KT models may lead to biased or misleading results when evaluated solely using a standardized benchmark.

To address these challenges, we propose several strategies. First, selective validation of datasets can be implemented, allowing researchers to identify and exclude problematic data points that could skew the results. Second, the benchmark itself can be adapted to better suit the specific context of the KT task, including modifications to the evaluation criteria or the introduction of additional metrics that capture the nuances of the dataset or learning environment. These strategies ensure that the benchmark remains relevant and accurate across diverse educational settings.

In practice, these considerations emphasize the importance of flexibility and adaptability when applying the unified benchmark in real-world educational scenarios. Researchers and practitioners must be aware of the limitations and be prepared to adjust their evaluation frameworks to ensure that they are capturing the true performance and effectiveness of KT models in their specific application contexts.

VIII. Future Research Directions

Even though state-of-the-art KT models have delivered promising results, the existing limitations and gaps in current methodologies and datasets present numerous potential directions for upcoming research. These future research directions can be guided by the GenKT framework proposed in this paper. For instance, the need for improved interpretability in KT models directly corresponds to the 'Student Knowledge Memory' and 'Auxiliary Knowledge Base' components of GenKT, where enhancing transparency and understanding of these models is critical. Similarly, addressing data sparsity and optimizing large-scale data processing are areas that align with the 'Data Representation' and 'Neural Network Design' components of the framework, suggesting targeted areas for future exploration and development within the GenKT structure.

Interpretability: Currently, a significant limitation of DLKT algorithms lies in their inherent lack of interpretability. This issue stems from the black-box nature of deep learning models, which makes it challenging to understand the underlying decision-making processes and the relationships between inputs and outputs. The lack of interpretability is not merely a theoretical concern but has practical implications, particularly in the educational domain. Transparency and credibility are critical factors for the acceptance and integration of technology in educational settings. Educators, students, and policymakers often require a clear understanding of how algorithms arrive at specific predictions or decisions to trust and effectively utilize these systems. This lack of transparency undermines confidence in the technology and poses a barrier to its widespread adoption, as stakeholders are unable to fully evaluate the reliability or fairness of the algorithms. For example, the deployment of AI teaching assistants, such as Georgia Tech's Jill Watson has raised concerns about the black-box nature of their decision-making processes, leading to unease among educators and students regarding the transparency and reliability of these systems [118]. Therefore, researching and developing highly interpretable DLKT algorithms is not only a solution to technical limitations but also a direct response to the current needs of educational technology research. Without a doubt, this will be an important direction and challenge for future research.

Nonbinary modeling: In the current body of research within the KT domain, the task is predominantly framed as a binary classification problem, wherein the primary objective is to predict the correctness of a student's response to a given question.

This formulation simplifies the complexity of student learning dynamics, treating all questions as having a uniform binary outcome—correct or incorrect. However, in real-world educational environments, students encounter a diverse range of question formats that go beyond binary responses. These include, but are not limited to, multiple-choice questions (where students select a single correct option), multiple-answer questions (requiring students to identify all correct options), fill-in-the-blank questions (demanding precise textual or numerical inputs), and computational problems (which may require stepwise reasoning or numerical calculations). Each of these question types presents distinct characteristics in terms of cognitive processes, difficulty levels, and answer formats.

Privacy-preserving KT: An important direction for future research in KT lies in addressing the challenge of training KT models while safeguarding student privacy. With the increasing reliance on data-driven approaches, concerns about data security and privacy protection have become paramount, particularly in educational settings where sensitive information about students is involved. Developing methods that ensure robust privacy-preserving mechanisms—such as federated learning, differential privacy, or secure multi-party computation—can enable KT models to be trained on decentralized or encrypted data without compromising individual privacy. By balancing the need for accurate modeling with ethical considerations around data usage, researchers can pave the way for more secure and widely accepted applications of KT systems in real-world educational environments.

Utilization of rich auxiliary information: The abundance and diversity of educational data highlight the complexity and dynamism of the learning process, making the field inherently data-intensive. Despite this richness, the datasets currently utilized for KT tasks predominantly focus on students' question-solving records. These datasets typically include a limited range of features, such as the questions posed to students, the associated KCs, and their binary responses (e.g., correct or incorrect). While such data has been instrumental in advancing KT algorithms, it captures only a narrow slice of the broader educational experience. The absence of additional contextual and auxiliary information in these datasets significantly constrains the depth and scope of student knowledge modeling. Auxiliary information, encompassing factors such as the timing of interactions, the emotional states of students, the difficulty level of questions, classroom dynamics, and even metadata about learning resources, holds a wealth of untapped potential.

Optimization Strategies for Large-Scale Data Processing: In large-scale data processing, optimizing performance is crucial for ensuring that KT models are both efficient and scalable. The following strategies can be employed to enhance the performance of KT models when dealing with vast amounts of educational data:

- *Parallel Computing and Distributed Systems:* Utilizing parallel computing frameworks such as GPUs and TPUs can significantly speed up the training and inference processes of KT models, particularly those employing attention-based mechanisms. Distributed computing frameworks like Apache Spark or distributed deep learning libraries can also be leveraged to handle data preprocessing and model training across multiple nodes, reducing the time required for large-scale data processing.
- *Model Pruning and Quantization:* Model pruning involves removing redundant or less important weights from the model, which can reduce the model's size and speed up computations. Quantization, which involves reducing the precision of the model's weights (e.g., from 32-bit floating-point to 8-bit integers), can also significantly decrease memory usage and increase processing speed, making it more feasible to deploy KT models in resource-constrained environments.
- *Efficient Data Handling Techniques:* Employing data augmentation and efficient batching strategies can optimize the way data is fed into KT models. Additionally, using advanced data preprocessing pipelines that include techniques like feature selection, normalization, and dimensionality reduction can help manage large datasets more effectively, leading to faster model training and inference times.
- *Scalable Attention Mechanisms:* For attention-based KT models, optimizing the attention mechanism itself can lead to significant performance gains. Techniques such as sparse attention, low-rank approximations, or adaptive attention can reduce the computational burden associated with large attention matrices, thereby enhancing scalability.

Addressing Data Sparsity: Despite the promising results achieved by state-of-the-art KT models, data sparsity remains a significant challenge that hinders the effectiveness and generalizability of these models. In educational datasets, it is common to encounter questions that appear only a few times or even just once, which can severely limit the model's ability to learn meaningful patterns. While approaches such as the 'Train on KC, Test on Question' paradigm offer partial relief from the effects of sparse data, they do not fully resolve the underlying issue. Recent advancements in contrastive learning, as demonstrated by the Bi-CLKT model [44] and others [98], [99], [100], provide promising avenues for addressing data sparsity in KT. These models enhance the differentiation of learned representations by contrasting positive and negative pairs, thereby improving predictive accuracy and robustness. Exploring and integrating these contrastive learning techniques into KT models could be a fruitful direction for future research, enabling more effective learning even in the face of sparse data.

IX. Conclusion

In this survey, we systematically summarize works in the field of KT. We provide a clear definition of KT tasks and also introduce their differences from IRT and CDMs. Besides, we propose a generalized modeling framework to divide the DLKT tasks into five components: multimodal data encoding, student knowledge memory, auxiliary knowledge base, learning outcome objective, and computational efficiency and scalability, and based on this framework, we also make a classified review of the existing relevant papers. More importantly, we propose a standardized DLKT benchmark platform, named pyKT, to

guarantee that various methods are compared equitably and transparently. With this platform, we conduct empirical studies to evaluate the performance of the current mainstream DLKT algorithms in a fair and transparent environment. Finally, we outline the applications of KT techniques in the education as well as their future development directions.

REFERENCES

[1] L. Tetzlaff, F. Schmiedek, and G. Brod, "Developing personalized education: A dynamic framework," *Educ. Psychol. Rev.*, vol. 33, pp. 863–882, 2021.

[2] R. Reber, E. A. Canning, and J. M. Harackiewicz, "Personalized education to increase interest," *Curr. Directions Psychol. Sci.*, vol. 27, no. 6, pp. 449–454, 2018.

[3] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 750–757.

[4] Y. Qiu, Y. Qi, H. Lu, Z. A. Pardos, and N. T. Heffernan, "Does time matter? modeling the effect of time with Bayesian knowledge tracing," in *Proc. 4th Int. Conf. Educ. Data Mining*, 2011, pp. 139–148.

[5] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Model. User-Adapted Interact.*, vol. 4, pp. 253–278, 1994.

[6] A. Ghosh, N. Heffernan, and A. S. Lan, "Context-aware attentive knowledge tracing," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2330–2339.

[7] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," 2019, *arXiv: 1907.06837*.

[8] C. Piech et al., "Deep knowledge tracing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 505–513.

[9] S. Shen et al., "Convolutional knowledge tracing: Modeling individualization in student learning process," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 1857–1860.

[10] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 765–774.

[11] J. Lee and D.-Y. Yeung, "Knowledge query network for knowledge tracing: How knowledge interacts with skills," in *Proc. 9th Int. Conf. Learn. Analytics Knowl.*, 2019, pp. 491–500.

[12] Q. Liu et al., "EKT: Exercise-aware knowledge tracing for student performance prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 100–115, Jan. 2021.

[13] M. Dai, J.-L. Hung, X. Du, H. Tang, and H. Li, "Knowledge tracing: A review of available techniques," *J. Educ. Technol. Develop. Exchange*, vol. 14, no. 2, pp. 1–20, 2021.

[14] X. Song, J. Li, T. Cai, S. Yang, T. Yang, and C. Liu, "A survey on deep learning based knowledge tracing," *Knowl. Based Syst.*, vol. 258, 2022, Art. no. 110036.

[15] G. Abdelrahman, Q. Wang, and B. Nunes, "Knowledge tracing: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–37, 2023.

[16] Q. Liu, S. Shen, Z. Huang, E. Chen, and Y. Zheng, "A survey of knowledge tracing," 2021, *arXiv:2105.15106*.

[17] F. B. Baker, *The Basics of Item Response Theory*. Bloomington, IN, USA: ERIC, 2001.

[18] G. Rasch, *Probabilistic Models for Some Intelligence and Attainment Tests*. Copenhagen, Denmark: Danish Inst. Educ. Res., 1960.

[19] F. Lord and M. Novick, *Statistical Theories of Mental Test Scores*. Addison-Wesley: Addison-Wesley, 1968.

[20] F. Samejima, "Estimation of latent ability using a response pattern of graded scores," *Psychometrika Monograph Suppl.*, vol. 34, pp. 1–97, 1969.

[21] E. Muraki, "A generalized partial credit model: Application of an EM algorithm," *Appl. Psychol. Meas.*, vol. 16, no. 2, pp. 159–176, 1992.

[22] J. P. Leighton and M. J. Gierl, *Cognitive Diagnostic Assessment for Education: Theory and Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[23] K. K. Tatsuoka, "Rule space: An approach for dealing with misconceptions based on item response theory," *J. Educ. Meas.*, vol. 20, pp. 345–354, 1983.

[24] K. K. Tatsuoka, *Cognitive Assessment: An Introduction to the Rule Space Method*. Evanston, IL, USA: Routledge, 2009.

[25] L. V. DiBello, L. A. Roussos, and W. Stout, "Review of cognitively diagnostic assessment and a summary of psychometric models," in *Handbook of Statistics: Psychometrics*, vol. 26, R. Rao and S. Sinharay, Eds. Amsterdam, The Netherlands: Elsevier, 2007, pp. 970–1030.

[26] A. A. Rupp and J. Templin, "Unique characteristics of diagnostic classification models: A comprehensive review of the current state-of-the-art," *Measurement*, vol. 6, pp. 219–262, 2008.

[27] J. de la Torre, "DINA model and parameter estimation: A didactic," *J. Educ. Behav. Statist.*, vol. 34, pp. 115–130, 2009.

[28] B. W. Junker and K. Sijtsma, "Cognitive assessment models with few assumptions, and connections with nonparametric item response theory," *Appl. Psychol. Meas.*, vol. 25, pp. 258–272, 2001.

[29] J. L. Templin and R. A. Henson, "Measurement of psychological disorders using cognitive diagnosis models," *Psychol. Methods*, vol. 11, 2006, Art. no. 287.

[30] J. de la Torre, "The generalized DINA model framework," *Psychometrika*, vol. 76, pp. 179–199, 2011.

[31] R. A. Henson, J. L. Templin, and J. T. Willse, "Defining a family of cognitive diagnosis models using log-linear models with latent variables," *Psychometrika*, vol. 74, 2009, Art. no. 191.

[32] S. Shen, Z. Huang, Q. Liu, Y. Su, S. Wang, and E. Chen, "Assessing student's dynamic knowledge state by exploring the question difficulty effect," in *Proc. 45th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2022, pp. 427–437.

[33] S. Sonkar, A. E. Waters, A. S. Lan, P. J. Grimaldi, and R. G. Baraniuk, "qDKT: Question-centric deep knowledge tracing," 2020, *arXiv: 2005.12442*.

[34] C.-K. Yeung, "Deep-IRT: Make deep learning based knowledge tracing explainable using item response theory," 2019, *arXiv: 1904.11738*.

[35] Z. Liu, Q. Liu, J. Chen, S. Huang, and W. Luo, "simpleKT: A simple but tough-to-beat baseline for knowledge tracing," 2023, *arXiv:2302.06881*.

[36] S. Huang, Z. Liu, X. Zhao, W. Luo, and J. Weng, "Towards robust knowledge tracing models via K-sparse attention," in *Proc. 46th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2023, pp. 2441–2445.

[37] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis –A new alternative to knowledge tracing," in *Proc. 2009 Conf. Artif. Intell. Educ.: Building Learn. Syst. Care: Knowl. Representation Affect. Modelling*, 2009, pp. 531–538.

[38] S. Shen et al., "Learning process-consistent knowledge tracing," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 2021, pp. 1452–1460.

[39] T. Guo et al., "Graduate employment prediction with bias," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 670–677.

[40] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: Lower dimensional target embedding," *Image Vis. Comput.*, vol. 75, pp. 21–31, 2018.

[41] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2019, Art. no. 721.

[42] S. Tong et al., "Structure-based knowledge tracing: An influence propagation view," in *Proc. 2020 IEEE Int. Conf. Data Mining*, 2020, pp. 541–550.

[43] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, "Graph-based knowledge tracing: Modeling student proficiency using graph neural network," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell.*, 2019, pp. 156–163.

[44] X. Song, J. Li, Q. Lei, W. Zhao, Y. Chen, and A. Mian, "Bi-CLKT: Bi-graph contrastive learning based knowledge tracing," *Knowl. Based Syst.*, vol. 241, 2022, Art. no. 108274.

[45] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, and Z. Guan, "JKT: A joint graph convolutional network based deep knowledge tracing," *Inf. Sci.*, vol. 580, pp. 510–523, 2021.

[46] S. Pu, M. Yudelson, L. Ou, and Y. Huang, "Deep knowledge tracing with transformers," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, Springer, 2020, pp. 252–256.

[47] X. Guo, Z. Huang, J. Gao, M. Shang, M. Shu, and J. Sun, "Enhancing knowledge tracing via adversarial training," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 367–375.

[48] Y. Choi et al., "Towards an appropriate query, key, and value computation for knowledge tracing," in *Proc. 7th ACM Conf. Learn. Scale*, 2020, pp. 341–344.

[49] D. Shin, Y. Shim, H. Yu, S. Lee, B. Kim, and Y. Choi, "SAINT+: Integrating temporal features for EdNet correctness prediction," in *Proc. 11th Int. Learn. Analytics Knowl. Conf.*, 2021, pp. 490–496.

[50] Y. Yang et al., "GIKT: A graph-based interaction model for knowledge tracing," in *Proc. Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, Springer, 2021, pp. 299–315.

[51] W.-L. Chan and D.-Y. Yeung, "Clickstream knowledge tracing: Modeling how students answer interactive online questions," in *Proc. 11th Int. Learn. Analytics Knowl. Conf.*, 2021, pp. 99–109.

[52] M. Zhang, X. Zhu, C. Zhang, Y. Ji, F. Pan, and C. Yin, "Multi-factors aware dual-attentional knowledge tracing," in *Proc. 30th ACM Int. Conf. Inf. Knowl. Manage.*, 2021, pp. 2588–2597.

[53] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma, "Augmenting knowledge tracing by considering forgetting behavior," in *Proc. World Wide Web Conf.*, 2019, pp. 3101–3107.

[54] Y. Liu, Y. Yang, X. Chen, J. Shen, H. Zhang, and Y. Yu, "Improving knowledge tracing via pre-training question embeddings," 2020, *arXiv: 2012.05031*.

[55] T. Long, Y. Liu, J. Shen, W. Zhang, and Y. Yu, "Tracing knowledge state with individual cognition and acquisition estimation," in *Proc. 44th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2021, pp. 173–182.

[56] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[57] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv: 1810.04805*.

[58] Y. Su et al., "Exercise-enhanced sequential modeling for student performance prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2018, Art. no. 297.

[59] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series," in *Proc. 2019 IEEE Int. Conf. Big Data*, 2019, pp. 3285–3292.

[60] S. Cheng et al., "AdaptKT: A domain adaptable method for knowledge tracing," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 123–131.

[61] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Trans. Assoc. Comput. Linguistics*, vol. 5, pp. 135–146, 2017.

[62] H. Tong, Y. Zhou, and Z. Wang, "Exercise hierarchical feature enhanced knowledge tracing," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, Springer, 2020, pp. 324–328.

[63] S. Arora, Y. Liang, and T. Ma, "A simple but tough-to-beat baseline for sentence embeddings," in *Proc. Int. Conf. Learn. Representations*, 2017, pp. 1–16.

[64] Y. Yin et al., "QuesNet: A unified representation for heterogeneous test questions," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1328–1336.

[65] T. Huang, M. Liang, H. Yang, Z. Li, T. Yu, and S. Hu, "Context-aware knowledge tracing integrated with the exercise representation and association in mathematics," *Int. Educ. Data Mining Soc*, 2021 pp. 360–366.

[66] Z. Wang, X. Feng, J. Tang, G. Y. Huang, and Z. Liu, "Deep knowledge tracing with side information," in *Proc. 20th Int. Conf. Artif. Intell. Educ.*, Springer, 2019, pp. 303–308.

[67] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "LINE: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[68] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[69] H. Tong, Z. Wang, Y. Zhou, S. Tong, W. Han, and Q. Liu, "HGKT: Introducing hierarchical exercise graph for knowledge tracing," 2020, *arXiv: 2006.16915*.

[70] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[71] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[72] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*.

[73] A. Graves and A. Graves, "Long short-term memory," in *Supervised Sequence Labelling With Recurrent Neural Networks*. Berlin, Germany: Springer, 2012, pp. 37–45.

[74] C.-K. Yeung and D.-Y. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proc. 5th Annu. ACM Conf. Learn. Scale*, 2018, pp. 1–10.

[75] Z. Huang et al., "Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students," *ACM Trans. Inf. Syst.*, vol. 38, no. 2, pp. 1–33, 2020.

[76] B. Ma, G. P. Hettiarachchi, S. Fukui, and Y. Ando, "Each encounter counts: Modeling language learning and forgetting," in *Proc. 13th Int. Learn. Analytics Knowl. Conf.*, 2023, pp. 79–88.

[77] B. Choffin, F. Popineau, Y. Bourda, and J.-J. Vie, "DAS3H: Modeling student learning and forgetting for optimally scheduling distributed practice of skills," in *Proc. Int. Conf. Educ. Data Mining*, 2019, pp. 1–10.

[78] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck, "Going deeper with deep knowledge tracing," *Int. Educ. Data Mining Soc.*, 2016 pp. 545–550.

[79] Y. Lu, D. Wang, Q. Meng, and P. Chen, "Towards interpretable deep learning models for knowledge tracing," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, Springer, 2020, pp. 185–190.

[80] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J.-J. Vie, "Deep knowledge tracing and dynamic student classification for knowledge tracing," in *Proc. 2018 IEEE Int. Conf. Data Mining*, 2018, pp. 1182–1187.

[81] J. Chen, Z. Liu, S. Huang, Q. Liu, and W. Luo, "Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations," 2023, *arXiv:2302.06885*.

[82] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," 2016, *arXiv:1606.03126*.

[83] G. Abdelrahman and Q. Wang, "Knowledge tracing with sequential key-value memory networks," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 175–184.

[84] F. Ai et al., "Concept-aware deep knowledge tracing and exercise recommendation in an online learning system," *Int. Educ. Data Mining Soc.*, 2019, pp. 175–184.

[85] J. Zhao, S. Bhatt, C. Thille, N. Gattani, and D. Zimmaro, "Cold start knowledge tracing with attentive neural turing machine," in *Proc. 7th ACM Conf. Learn. Scale*, 2020, pp. 333–336.

[86] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[87] S. Pandey and J. Srivastava, "RKT: Relation-aware self-attention for knowledge tracing," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, 2020, pp. 1205–1214.

[88] Y. Zhou, X. Li, Y. Cao, X. Zhao, Q. Ye, and J. Lv, "LANA: Towards personalized deep knowledge tracing through distinguishable interactive sequences," 2021, *arXiv:2105.06266*.

[89] T. Long et al., "Improving knowledge tracing with collaborative information," in *Proc. 15th ACM Int. Conf. Web Search Data Mining*, 2022, pp. 599–607.

[90] A. G. Hawkes, "Spectra of some self-exciting and mutually exciting point processes," *Biometrika*, vol. 58, no. 1, pp. 83–90, 1971.

[91] C. Wang et al., "Temporal cross-effects in knowledge tracing," in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, 2021, pp. 517–525.

[92] G. Abdelrahman and Q. Wang, "Deep graph memory networks for forgetting-robust knowledge tracing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 7844–7855, Aug. 2023.

[93] L. Zhang, X. Xiong, S. Zhao, A. Botelho, and N. T. Heffernan, "Incorporating rich features into deep knowledge tracing," in *Proc. 4th ACM Conf. Learn. Scale*, 2017, pp. 169–172.

[94] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian, "Prerequisite-driven deep knowledge tracing," in *Proc. 2018 IEEE Int. Conf. Data Mining*, 2018, pp. 39–48.

[95] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," in *Proc. Int. Conf. Learn. Representations*, 2016, pp. 1–13.

[96] A. Ghosh, J. Raspat, and A. Lan, "Option tracing: Beyond correctness analysis in knowledge tracing," in *Proc. Int. Conf. Artif. Intell. Educ.*, Springer, 2021, pp. 137–149.

[97] Z. Liu, Q. Liu, J. Chen, S. Huang, J. Tang, and W. Luo, "pyKT: A python library to benchmark deep learning based knowledge tracing models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2022, pp. 18542–18555.

[98] W. Lee, J. Chun, Y. Lee, K. Park, and S. Park, "Contrastive learning for knowledge tracing," in *Proc. ACM Web Conf.*, 2022, pp. 2330–2338.

[99] Y. Zhao, H. Ma, J. Wang, X. He, and L. Chang, "Question-response representation with dual-level contrastive learning for improving knowledge tracing," *Inf. Sci.*, vol. 658, 2024, Art. no. 120032.

[100] Y. Yin et al., "Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer," in *Proc. ACM Web Conf.*, 2023, pp. 855–864.

[101] Z. Liu et al., "Enhancing deep knowledge tracing with auxiliary tasks," in *Proc. World Wide Web Conf.*, 2023, pp. 4178–4187.

[102] S. Oxman, W. Wong, and D. Innovations, "White paper: Adaptive learning systems," 2014.

[103] S. Shehata, D. S. Calonge, P. Purnell, and M. Thompson, "Enhancing video-based learning using knowledge tracing: Personalizing students' learning experience with orbits," in *Proc. 18th Workshop Innov. Use NLP Building Educ. Appl.*, 2023, pp. 100–107.

[104] Wikipedia contributors, "Intelligent tutoring system — Wikipedia, the free encyclopedia," 2025. Accessed: Jan. 14, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Intelligent_tutoring_system
[105] V. J. Shute, M. Ventura, and Y. J. Kim, "Assessment and learning of qualitative physics in Newton's playground," *J. Educ. Res.*, vol. 106, no. 6, pp. 423–430, 2013.
[106] Y. Cui, M.-W. Chu, and F. Chen, "Analyzing student process data in game-based assessments with Bayesian knowledge tracing and dynamic Bayesian networks," *J. Educ. Data Mining*, vol. 11, no. 1, pp. 80–100, Jun. 2019.
[107] P. Kantharaju, K. Alderfer, J. Zhu, B. Char, B. Smith, and S. Ontanon, "Tracing player knowledge in a parallel programming educational game," in *Proc. AAAI Conf. Artif. Intell. Interactive Digit. Entertainment*, 2018, pp. 173–179. [Online]. Available: https://ojs.aaai.org/index.php/AIIDE/article/view/13038
[108] W. Ma, O. O. Adesope, J. C. Nesbit, and Q. Liu, "Intelligent tutoring systems and learning outcomes: A meta-analysis," *J. Educ. Psychol.*, vol. 106, no. 4, 2014, Art. no. 901.
[109] R. S. D. Baker et al., "Contextual slip and prediction of student performance after use of an intelligent tutor," in *Proc. 18th Int. Conf. User Model. Adapt. Personalization*, Springer, 2010, pp. 52–63.
[110] Y. Mao, "Deep learning vs. Bayesian knowledge tracing: Student models for interventions," *J. Educ. Data Mining*, vol. 10, no. 2, pp. 28–54, 2018.
[111] J. A. Marsh and C. C. Farrell, "How leaders can support teachers with data-driven decision making: A framework for understanding capacity building," *Educ. Manage. Admin. Leadership*, vol. 43, no. 2, pp. 269–289, 2015.
[112] C. Fischer et al., "Mining big data in education: Affordances and challenges," *Rev. Res. Educ.*, vol. 44, no. 1, pp. 130–160, 2020.
[113] C. Romero and S. Ventura, "Data mining in education," *WIREs Data Mining Knowl. Discov.*, vol. 3, no. 1, pp. 12–27, 2013.
[114] Y. Lu, D. Wang, Q. Meng, and P. Chen, "Towards interpretable deep learning models for knowledge tracing," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, 2020, pp. 185–190.
[115] X. Ding and E. C. Larson, "On the interpretability of deep learning based models for knowledge tracing," 2021, *arXiv:2101.11335*.
[116] H. Zhang, Z. Liu, C. Shang, D. Li, and Y. Jiang, "A question-centric multi-experts contrastive learning framework for improving the accuracy and interpretability of deep sequential knowledge tracing models," 2024, *arXiv:2403.07322*.
[117] J. Chen, Z. Liu, S. Huang, Q. Liu, and W. Luo, "Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 14196–14204.
[118] W. Contributors, "Intelligent tutoring system — Wikipedia, the free Encyclopedia," 2025. Accessed: Jan. 14, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Intelligent_tutoring_system

Zitao Liu (Member, IEEE) is the dean with the Guangdong Institute of Smart Education, Jinan University, Guangzhou, China. His research is in the area of machine learning, and includes contributions in the areas of artificial intelligence in education and educational data mining. He has published more than 80 papers in highly ranked conference proceedings, such as NeurIPS, CVPR, AAAI, WWW, etc. and his applied research has resulted in more than 40 technology transfer and patents. He serves as the Executive Committee of the International AI in Education Society and the program co-chairs of the 25th International Conference on Artificial Intelligence in Education (AIED 2024).

Teng Guo received the PhD degree from the Dalian University of Technology. He is an assistant professor with the Guangdong Institute of Smart Education, Jinan University, Guangzhou, China. His research interests include data mining, smart education, and large language models. He has published more than 20 papers in highly-ranked conference proceedings and journals, including AAAI, NeurIPS, and WWW, etc. He serves as the Vitual Experience co-chairs of the 25th International Conference on Artificial Intelligence in Education (AIED 2024).

Qianru Liang is an assistant professor with the Guangdong Institute of Smart Education, Jinan University, Guangzhou, China. Her research interests include educational assessment, psychometric models, cognitive diagnosis models, digital literacy, 21st century skills, and student well-being. She has published papers in educational journals and conferences such as the *Computers in Human Behavior*, *Journal of Educational and Behavioral Statistics*, AERA, and NCME. Notable projects in her portfolio include assessing students' digital literacy learning trajectories and exploring strategies/interventions to enhance students' well-being in real-world and digital environments.

Mingliang Hou received the MSc degree from Shandong University, and the PhD degree from the Dalian University of Technology, China. He is currently a joint training postdoctoral researcher with the Guangdong Institute of Smart Education, Jinan University and the TAL Education Group. His research interests include smart education, social computing, and large language models.

Bojun Zhan is currently working toward the master's degree with the School of Cyberspace Security, Jinan University, Guangzhou, China. His research interests include artificial intelligence in education, knowledge tracing, and natural language processing.

Jiliang Tang received the PhD degree from Arizona State University, in 2015. He is an associate professor with the Computer Science and Engineering Department, Michigan State University. Before that, he was a research scientist with Yahoo Research. His research focuses on data mining, machine learning and their applications on social, web, and education domains. He was the recipient of the 2021 ACSIC Rising Star Award, 2021 IEEE Big Data Security Junior Research Award, 2020 ACM SIGKDD Rising Star Award, 2020 Distinguished Withrow Research Award, 2019 NSF Career Award and 7 best paper (or runner up) awards including KDD2015 and WSDM2018. His dissertation won the 2015 KDD Best Dissertation runner-up and Dean's Dissertation Award. He has served as the editors and the organizers in prestigious journals (e.g., the *ACM Transactions on Knowledge Discovery from Data*) and conferences (e.g., KDD, WSDM, and SDM). He has filed more than 10 US patents and has published his research in highly ranked journals and top conference proceedings, which received more than 30,000 citations with h-index 88 and extensive media coverage. More details about him can be found at https://www.cse.msu.edu/~tangjili/.

Weiqi Luo received the BS degree from Jinan University, Guangzhou, in 1982, and the PhD degree from the South China University of Technology, in 2000. Currently, he is a professor with the School of Information Science and Technology, Guangdong Institute of Smart Education, Jinan University. His research interests include network security, artificial intelligence, smart education, and Big Data. He has published more than 100 high-quality papers in international journals and conferences.

Jian Weng received the BS and MS degrees from the South China University of Technology, in 2001 and 2004, respectively, and the PhD degree from Shanghai Jiao Tong University, in 2008. He is a professor and the executive dean with the College of Information Science and Technology, Jinan University. His research areas include public key cryptography, cloud security, blockchain, etc. He has published 80 papers in international conferences and journals such as CRYPTO, EUROCRYPT, ASIACRYPT, TCC, PKC, CT-RSA, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *IEEE Transactions on Dependable and Secure Computing*, etc. He also serves as associate editor of *IEEE Transactions on Vehicular Technology*. He received the Ding Ying Science and Technology Award from Guangdong Province in 2023. He received the Young Scientists Fund of the National Natural Science Foundation of China in 2018, and the Cryptography Innovation Award from Chinese Association for Cryptologic Research (CACR) in 2015.