A Survey of Knowledge Tracing: Models, Variants, and Applications

Shuanghong Shen, Qi Liu, *Member, IEEE*, Zhenya Huang, *Member, IEEE*, Yonghe Zheng, Minghao Yin, *Member, IEEE*, Minjuan Wang, *Member, IEEE*, and Enhong Chen, *Fellow, IEEE*

(Survey Paper)

***Abstract*—Modern online education has the capacity to provide intelligent educational services by automatically analyzing substantial amounts of student behavioral data. Knowledge tracing (KT) is one of the fundamental tasks for student behavioral data analysis, aiming to monitor students' evolving knowledge state during their problem-solving process. In recent years, a substantial number of studies have concentrated on this rapidly growing field, significantly contributing to its advancements. In this survey, we will conduct a thorough investigation of these progressions. First, we present three types of fundamental KT models with distinct technical routes. Subsequently, we review extensive variants of the fundamental KT models that consider more stringent learning assumptions. Moreover, the development of KT cannot be separated from its applications, so we present typical KT applications in various scenarios. To facilitate the work of researchers and practitioners in this field, we have developed two open-source algorithm libraries: EduData that enables the downloading and preprocessing of KT-related datasets, and EduKTM that provides an extensible and unified implementation of existing mainstream KT models. Finally, we discuss potential directions for future research in this rapidly growing field. We hope that the current survey will assist both researchers and practitioners in fostering the development of KT, thereby benefiting a broader range of students.**

***Index Terms*—Adaptive learning, educational data mining, knowledge tracing (KT), online education, user modeling.**

I. Introduction

With the proliferation of the Internet and mobile communication technology, online education has become increasingly popular and is now developing at an unprecedented scale [1], [2], [3]. This innovative style of learning provides a degree of flexibility that conventional education cannot match, which enables teaching and learning to occur at any time and any place. Meanwhile, online learning systems (e.g., Coursera, ASSISTment) [4], [5], [6] have proven to be even more effective than traditional learning styles, since they can offer more intelligent educational services, such as recommending individualized learning resources to students [7], [8]. To provide these intelligent services, online learning systems continuously record a massive amount of available data about student–system interactions (e.g., responding to exercises), which can be further mined to assess their knowledge levels, learning preferences, and other attributes. Specifically, knowledge tracing (KT) [9] is one of the most fundamental and critical tasks for analyzing students' learning behavior data, which aims to explore the recorded student–system interactions to monitor their evolving knowledge states [10], [11].

Fig. 1 presents a simple schematic diagram of KT. During the learning process, online learning systems continuously record students' learning behavioral data, including exercises and their related knowledge concepts (e.g., *equality, inequality, plane vector*, and *probability*, represented in various colors), and students' answers (i.e., correct or incorrect responses). A substantial amount of supplementary information is also simultaneously recorded, including response time, opportunity count, and tutor intervention, which provides a more comprehensive reflection of students' learning process. Based on the collected learning data, researchers are striving to maintain an estimate of students' evolving knowledge states. For illustration, we give a case in Fig. 1, where the student's prior knowledge is quantified as 0.2, 0.4, 0.4, and 0.5 across four distinct knowledge concepts. The radar map serves as a visual representation of the student's knowledge mastery, which progressively expands as the student continues to acquire new knowledge in learning. After a period of learning, the student's knowledge states reach 0.9, 0.8, 0.8, and 0.7, respectively, suggesting good knowledge growth. In the aforementioned learning process, KT models aim to monitor changes in students' knowledge states. Once we

Manuscript received 23 November 2023; revised 2 February 2024; accepted 24 March 2024. Date of publication 8 April 2024; date of current version 9 July 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFF0901003, and in part by the National Natural Science Foundation of China under Grant 62337001 and Grant U20A20229. *(Corresponding authors: Enhong Chen; Qi Liu.)*

Shuanghong Shen and Enhong Chen are with the School of Data Science, University of Science and Technology of China, Hefei 230026, China, and also with the State Key Laboratory of Cognitive Intelligence, Hefei, Anhui 230026, China (e-mail: closer@mail.ustc.edu.cn; cheneh@ustc.edu.cn).

Qi Liu and Zhenya Huang are with the School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China, and also with the State Key Laboratory of Cognitive Intelligence and Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, Anhui 230026, China (e-mail: qiliuql@ustc.edu.cn; huangzhy@ustc.edu.cn).

Yonghe Zheng is with the Research Institute of Science Education, Faculty of Education, Beijing Normal University, Haidian, Beijing 100875, China (e-mail: zhengyonghe@bnu.edu.cn).

Minghao Yin is with the School of Information Science and Technology, Northeast Normal University, Changchun, Jilin 130117, China (e-mail: ymh@nenu.edu.cn).

Minjuan Wang is with The Education University of Hong Kong,, Hong Kong SAR 999077, China, and also with San Diego State University, San Diego, CA 92182 USA (e-mail: mwang@sdsu.edu).

Digital Object Identifier 10.1109/TLT.2024.3383325

1939-1382 © 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

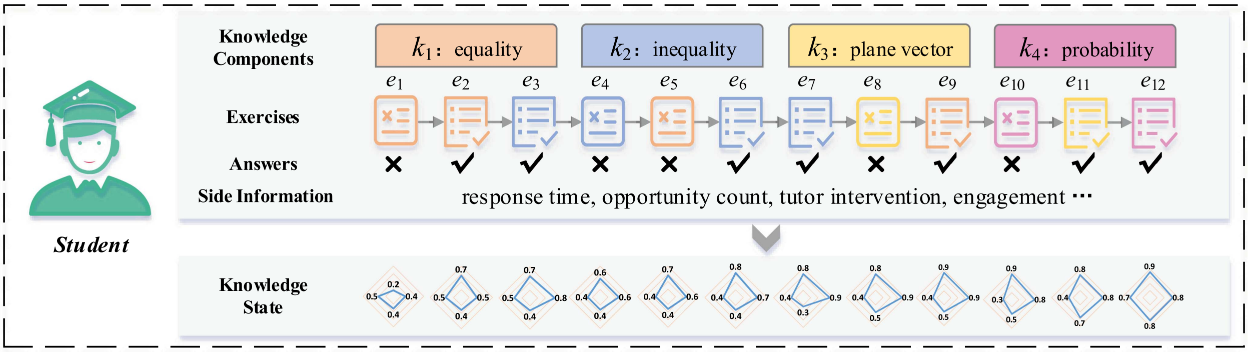

Fig. 1. Simple schematic diagram of KT. Different knowledge concepts are represented in different colors, while exercises are also depicted in the color relevant to the knowledge concepts. During the learning process, different kinds of side information are also recorded. The evolving process of the knowledge state is assessed by KT models and illustrated by the radar maps.

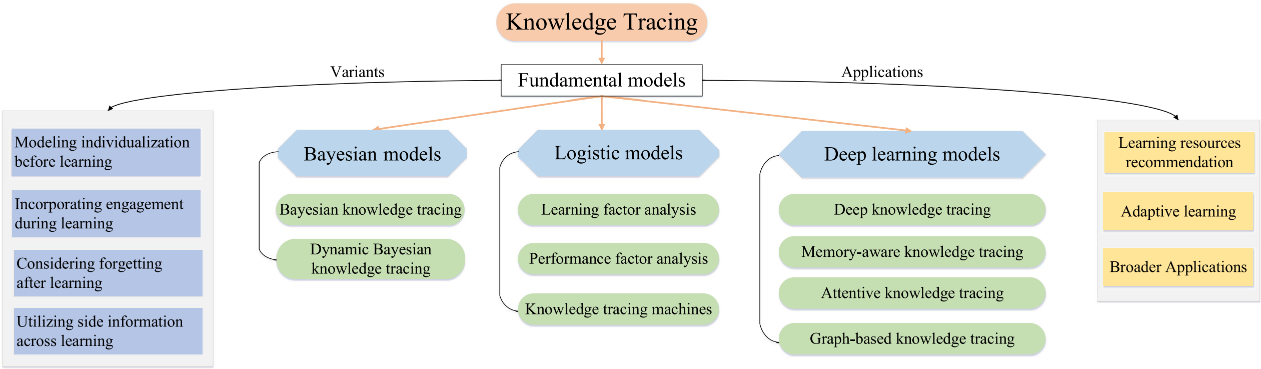
Knowledge Tracing
Fundamental models
Variants
Applications
Modeling individualization before learning
Incorporating engagement during learning
Considering forgetting after learning
Utilizing side information across learning
Bayesian models
Bayesian knowledge tracing
Dynamic Bayesian knowledge tracing
Logistic models
Learning factor analysis
Performance factor analysis
Knowledge tracing machines
Deep learning models
Deep knowledge tracing
Memory-aware knowledge tracing
Attentive knowledge tracing
Graph-based knowledge tracing
Learning resources recommendation
Adaptive learning
Broader Applications

Fig. 2. Overview of KT models.

understand students' knowledge states, the learning system can customize more suitable learning schemes for different students, thereby enabling the teaching of students in accordance with their proficiency. It also allows students to better comprehend their learning process and gradually focus on improving their skills with poorly mastered concepts [12], [13].

KT has been studied for decades, with the first studies tracing back to the late 1970s. These initial works primarily focused on confirming the effectiveness of mastery learning [14]. To the best of the authors' knowledge, Corbett and Anderson [9] were the first to introduce the concept of KT, employing Bayesian networks to model the student learning process, which they referred to as Bayesian knowledge tracing (BKT). Since then, the significance of KT has been recognized by a broader spectrum of researchers, and increasing attention has been directed toward KT-related research. Many logistic models have been applied to KT, including learning factor analysis (LFA) [15] and performance factor analysis (PFA) [16]. In recent years, deep learning has greatly enhanced research into the KT task, largely due to its capacity to extract and represent features and discover intricate structure. For instance, deep knowledge tracing (DKT) introduced recurrent neural networks (RNNs) [17] into the KT task and was found to significantly outperform previous methods [18]. Following this, various methods have been introduced that employ various types of neural networks to the KT task, considering various characteristics of the learning sequence [19], [20], [21]. Moreover, due to the requirements of practical applications, many variants of KT models have been continuously developed, and KT has already been broadly applied in numerous educational scenarios.

While novel KT models continue to emerge, there remains a lack of comprehensive surveys exploring this young research field, particularly regarding its numerous variants and applications. To this end, the current survey aims to systematically review the development of KT. As depicted in Fig. 2, we initially categorize existing KT models from a technical perspective, which is consistent with the majority of existing surveys [22], [23]. This categorization splits them into the following three categories:

1) Bayesian models;
2) logistic models;
3) deep learning models.

In each category, we further organize specific KT methods according to their various techniques. Subsequently, we introduce extensive variants of these fundamental KT models, which consider more stringent assumptions about more complete learning process in different learning phases. In addition, we present several typical applications of KT in real-learning scenarios. Due to the complexity of different KT models, we have open sourced two algorithm libraries to better aid researchers and practitioners in implementing KT models and facilitate community development in this domain. These libraries, EduData[1] and

[1] [Online]. Available: https://github.com/bigdata-ustc/EduData

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

EduKTM,[2] include most existing KT-related datasets, extensible and unified implementations of existing KT models, and relevant resources. Finally, we discuss potential future research directions. In summary, this article presents an extensive survey of KT that can serve as a comprehensive guide for both researchers and practitioners.

The rest of this article is organized as follows. Section II presents an overview of the KT task and we discuss the differences between this and previous surveys. Section III provides a review of the three categories of fundamental KT models. Section IV describes the variants of fundamental KT models. Section V introduces the extensive applications of KT in different scenarios. Section VI gives the summary of existing datasets for evaluating KT models and details of the algorithm libraries we have released. Section VII discusses some potential future research directions. Finally, Section VIII concludes this article.

II. Overview

A. Problem Definition

In an online learning system, suppose that there exists a set of students $\mathbb{S}$ and a set of exercises $\mathbb{E}$. Each exercise is related to specific knowledge concepts (KCs). Generally, the name given to the knowledge related to exercises differs across online learning platforms. For instance, it is named *skill* in ASSISTments [24]. To promote better understanding, we refer to these uniformly as knowledge concepts throughout this article, and denote the set of all KCs as $\mathbb{KC}$. Moreover, M and K are, respectively, used to represent the total number of exercises and KCs. Students are asked to answer different exercises in order to achieve mastery of the related knowledge. Therefore, the learning sequence of a student can be formulated as $\boldsymbol{X} = \{([e_1, k_{e_1}], a_1, r_1), ([e_2, k_{e_2}], a_2, r_2), \ldots, ([e_t, k_{e_t}], a_t, r_t), \ldots, ([e_N, k_{e_N}], a_N, r_N)\}$, where the tuple $([e_t, k_{e_t}], a_t, r_t)$ represents the learning interaction at the tth time step, e_t represents the exercise, k_{e_t} represents the exercise's related KCs, a_t represents the correctness label (i.e., with 1 for correct and 0 for incorrect answers), r_t stands for the side information recorded in this learning interaction, and N is the length of the learning sequence. The research problem of KT can thus be defined as follows.

Given sequences of learning interactions in online learning systems, KT aims to monitor students' evolving knowledge states during the learning process and predict their performance on future exercises. The measured knowledge states can be further applied to individualize students' learning schemes in order to maximize their learning efficiency.

Some recent works directly regarded the KT task as student performance prediction, without considering students' knowledge states [25], [26], [27]. We agree that predicting student performance is of great significance, as it is now the best way to evaluate the quality of the knowledge state traced by KT models. However, we have to point out that KT focuses more on students' knowledge states, especially their interpretability and rationality, which is related to the students' acceptance of the given conclusions based on the KT model [28], [29].

B. Categorization

As shown in Fig. 2, we categorize and summarize the existing KT models according to their technical differences. In more detail, the proposed taxonomy splits existing KT methods into the following three categories:

1) Bayesian models, which are implemented through probability model;
2) logistic models, which are implemented through logistic functions;
3) deep learning models, which are implemented through neural networks.

Specifically, we divide the deep learning models into four subcategories according to four various neural networks, i.e., DKT based on RNNs, memory-aware KT based on memory networks, attentive KT (AKT) based on self-attention mechanism, and graph-based KT (GKT) based on graph neural networks. In addition to these fundamental KT models, we also introduce a large number of their variants, which, respectively, consider a complete learning process in distinct learning phases, as follows:

1) modeling individualization before learning;
2) incorporating engagement during learning;
3) considering forgetting after learning;
4) utilizing side information across learning.

Moreover, we also summarize the extensive applications of KT in different educational scenarios, including learning resources recommendation, adaptive learning, and broader applications beyond student learning.

C. Differences Between This and Previous Surveys

Given the increasing importance of KT, several recent surveys have also examined this area. These include works by the authors in [22], [23], [30], [31], [32], and [33]. Here, we will briefly discuss the key distinctions between these studies to highlight the necessity and significance of this survey.

Existing surveys have either focused on specific categories of KT models or comprehensively reviewed all available KT models. For example, Pelánek [30] provided an overview of KT in terms of Bayesian models and logistic models, Song et al. [31] compared and discussed deep learning-based KT models, while Zanellati et al. [33] presented to pay more attention to hybrid models in KT. Liu [32] conducted a bibliometric analysis to examine the evolution of KT research from 1992 to 2021. Abdelrahman et al. [23] also presented a comprehensive survey for the KT literature, including a broad range of methods starting from the early attempts to the recent state-of-the-art techniques utilizing deep learning. Schmucker et al. [22] summarized KT methods in the context of student performance modeling problems.

However, current surveys are somewhat limited in their scope, they only provide a detailed introduction to various KT methods and comparisons between them. Given the complexity of online learning systems and the significant importance of KT research

[2] [Online]. Available: https://github.com/bigdata-ustc/EduKTM

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SUMMARY OF DIFFERENT TYPES OF FUNDAMENTAL KT MODELS

Category	Typical approach	Technique	KC relationship	Knowledge state
Bayesian models	BKT [9]	Bayesian networks	independent	unobservable node in HMM
	DBKT [35]	dynamic Bayesian networks	pre-defined	
Logistic models	LFA [15]	logistic regression	independent	the output of logistic regression function
	PFA [16]			
	KTM [36]	FMs		
Deep learning models	DKT [18]	RNN/LSTM	discover automatically	the hidden state
	Memory-aware KT [37]	memory networks	correlation weights	*value* matrix
	AKT [20], [21]	self-attention mechanism	attention weights	attentive historical knowledge state
	GKT [19]	graph neural networks	edges in graph	aggregate in the graph

in practical applications, this survey places a greater emphasis on the variants and applications of KT models, rather than solely introducing and comparing different KT methods. Moreover, considering that datasets are collected from different systems with various setting, subjects, learning stages, and scales, we do not report and compare the performance of KT models on the student performance prediction task across various datasets in this survey. Pavlik et al. [34] also empirically verified that no single KT model was always the best, a specific better model must consider multiple student features and the learning context. Instead, we have open sourced two algorithm libraries, which include the majority of existing KT-related datasets and unified implementations of existing KT models. Consequently, researchers and practitioners can freely select appropriate KT models based on their specific requirements in various application scenarios.

III. Fundamental KT Models

In this section, as given in Table I, we will present the fundamental KT models, based on our taxonomic framework. Specifically, we will introduce these models in accordance with their development timeline. Subsequently, we will provide a summary of these fundamental KT models.

A. Bayesian Models

Bayesian models assume that the learning process adheres to a Markov process. This process allows for the estimation of students' latent knowledge states based on their observed performance. In the subsequent section, we will present two Bayesian models in our taxonomy framework: the BKT and the dynamic BKT (DBKT).

1) Bayesian KT: BKT's structure is illustrated in Fig. 3; here, the unshaded nodes represent unobservable latent knowledge states, while the shaded nodes represent the observable answers of the student.

BKT is a unique instance of the hidden Markov model (HMM). There are two types of parameters in HMM: transition probabilities and emission probabilities. In BKT, the transition probabilities are defined by two learning parameters: 1) $P(T)$, the probability of transition from the unlearned state to the learned state and 2) $P(F)$, the probability of forgetting previously mastered knowledge. Moreover, the emission probabilities are determined by two performance parameters: 1) $P(G)$—the probability that a student will guess correctly, despite nonmastery and 2) $P(S)$—the probability that a student will make a mistake, despite mastery. in addition, $P(L_0)$ represents the initial probability of mastery. BKT operates within a two-state student modeling framework: knowledge is either learned or unlearned, and there is no forgetting once a student has mastered the knowledge. Based on the observations of the student's learning interactions, the following equation is utilized to estimate the knowledge state and the probability of correct answers:

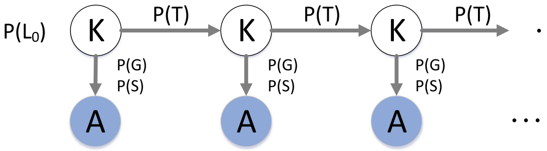

Fig. 3. Topology of BKT [9]. K are the unobserved knowledge nodes, A are the observed performance (answer) nodes, $P(L_0)$ represents the initial probability, $P(T)$ is the transition probability, $P(G)$ is the guessing probability, and $P(S)$ is the slipping probability.

$$P(L_n) = P(L_n|\text{Answer}) + (1 - P(L_n|\text{Answer}))P(T)$$
$$P(C_{n+1}) = P(L_n)(1 - P(S)) + (1 - P(L_n))P(G) \quad (1)$$

where $P(L_n)$ is the probability that a KC is mastered at the nth learning interaction, and $P(C_{n+1})$ is the probability of correct answers at the next learning interaction. $P(L_n)$ is the sum of two probabilities: 1) the probability that the KC is already mastered and 2) the probability that the knowledge state will convert to the mastered state. The posterior probability $P(L_n|\text{Answer})$ is estimated as follows:

$$P(L_n|\text{correct}) = \frac{P(L_{n-1})(1 - P(S))}{P(L_{n-1})(1 - P(S)) + (1 - P(L_{n-1}))P(G)}$$
$$P(L_n|\text{incorrect}) = \frac{P(L_{n-1})P(S)}{P(L_{n-1})P(S) + (1 - P(L_{n-1}))(1 - P(G))}. \quad (2)$$

2) Dynamic BKT: BKT independently models the parameters of each KC, employing a specific model for each KC.

However, KCs are not completely independent, but rather hierarchical and closely related [38]. Dynamic Bayesian networks are capable of jointly representing multiple skills within a single model, potentially enhancing the representational power of BKT. Consequently, Käser et al. [35] proposed DBKT to represent the hierarchies and relationships within KCs using dynamic Bayesian networks. This approach considers different KCs jointly within a single model.

In DBKT, a student's knowledge mastery is represented by binary latent variables, which are estimated based on their learning interactions. Unlike BKT, DBKT takes into account the dependencies between various KCs. For example, if KC_1 and KC_2 are prerequisites for mastering KC_3, students' mastery of KC_3 depends on their mastery of KC_1 and KC_2. Let H denote the unobserved variables, i.e., lack of student answers and binary mastery variables. Assuming that the student correctly answers an exercise associated with KC_1 at time step t_1, i.e., $a_{1,1} = 1$. The observed variables are, then, $a_m = a_{1,1}$, and the unobserved variables are $h_m = \{\text{KC}_{1,1}, \text{KC}_{2,1}, \text{KC}_{3,1}, a_{2,1}, a_{3,1}\}$. The objective of DBKT is to find the parameters θ that maximize the joint probability $p(a_m, h_m|\theta)$. The log-likelihood can alternatively be formulated using a log-linear model, as follows:

$$L(\boldsymbol{w}) = \sum_m ln\left(\sum_{h_m} \exp\left(\boldsymbol{w}^T \Phi(a_m, h_m) - \ln(Z)\right)\right) \quad (3)$$

where $\Phi : A \times H \rightarrow \mathbb{R}^F$ denotes a mapping from the observed space A and the latent space H to an F-dimensional feature vector. Z is a normalizing constant and $\boldsymbol{w}$ denotes the weights.

B. Logistic Models

Logistic models represent the probability of students correctly answering exercises as a logistic function of the student and KC parameters. They first use different factors in students' learning interactions to compute an estimation of the student and KC parameters, then utilize a logistic function to transform this estimation into the prediction of the probability of mastery [30]. In the subsequent section, we will introduce the following three types of logistic models:

1) LFA;
2) PFA;
3) Knowledge tracing machines (KTM).

1) Learning Factor Analysis: The LFA model [15] considers the following learning factors.

1) *Initial knowledge state:* Parameter α estimates the initial knowledge state of each student.
2) *Easiness of KCs:* Parameter β captures the easiness of different KCs.
3) *Learning rate of KCs:* Parameter γ denotes the learning rate of KCs.

The standard LFA model takes the following form:

$$p(\theta) = \sigma\left(\sum_{i \in N} \alpha_i S_i + \sum_{j \in \text{KC}s} (\beta_j + \gamma_j T_j)\, K_j\right) \quad (4)$$

where σ is the sigmoid function, S_i is the covariate for the student i, T_j represents the covariate for the number of interactions on KC j, K_j is the covariate for KC j, and $p(\theta)$ is the estimation of the probability of a correct answer.

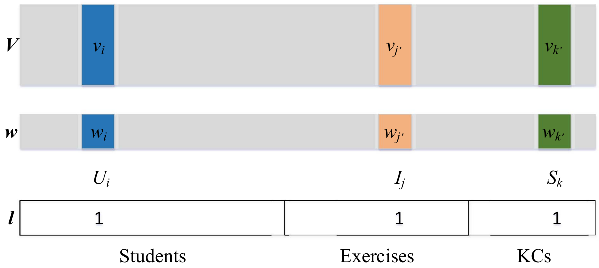

Fig. 4. Example of activation of a KTM [36]. V refers to the matrix of embeddings, w refers to the vector of biases, x is the encoding vector of the learning interaction.

2) Performance Factor Analysis: The PFA model [16] can be seen as an extension of the LFA model that is especially sensitive to the student performance. In contrast to the LFA model, PFA considers the following different factors.

1) *Previous failures:* Parameter f is the prior failures for the KC of the student.
2) *Previous successes:* Parameter s represents the prior successes for the KC of the student.
3) *Easiness of KCs:* Parameter β means the easiness of different KCs, which is the same as in the LFA model.

The standard PFA model takes the following form:

$$p(\theta) = \sigma\left(\sum_{j \in \text{KC}s} (\beta_j + \mu_j s_{ij} + \nu_j f_{ij})\right) \quad (5)$$

where μ and ν are the coefficients for s and f, which denote the learning rates for successes and failures, respectively.

3) KT Machines: The KTM model, developed by Vie and Kashima [36], employs factorization machines (FMs) [39], [40] to generalize logistic models to higher dimensions. FMs were initially introduced as a general predictor capable of working with any real-valued feature vector, enabling the model to represent all interactions between variables using factorized parameters [41]. FMs provide a means of encoding side information about exercises or students into the model. Fig. 4 illustrates the example of KTM, which models the knowledge mastery of the student based on a sparse set of weights for all features involved in the event. Let L be the number of features; here, the features can be related to students, exercises, KCs, or any other side information. The learning interaction is encoded by a sparse vector $\boldsymbol{l}$ of length L. When feature i is involved in the interaction, $l_i > 0$, the probability $p(\theta)$ of the correct answer is determined by the following equation:

$$p(\theta) = \sigma\left(\mu + \sum_{i=1}^{L} w_i l_i + \sum_{1 \le i < j \le L} l_i l_j \left\langle \boldsymbol{v_i}, \boldsymbol{v_j} \right\rangle\right) \quad (6)$$

where μ is the global bias, and the feature i is modeled by the bias $w_i \in \boldsymbol{R}$ and the embedding $\boldsymbol{v}_i \in \boldsymbol{R}^d$ (d is the dimension). Note that only features with $l_i > 0$ will have impacts on the predictions.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

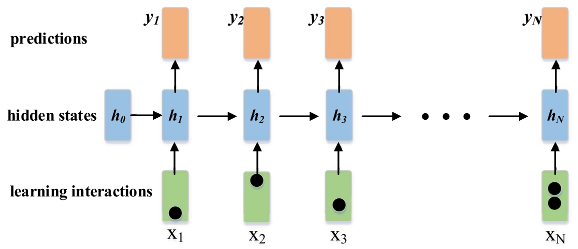

Fig. 5. Architecture of DKT [18].

C. Deep Learning Models

The cognitive process can be influenced by various factors at both the macro and microlevels. It is difficult for Bayesian models or logistic models to adequately capture a cognitive process of high complexity [18]. Deep learning, with its potent ability to achieve nonlinearity and feature extraction, is well suited for modeling complex learning processes, particularly when a significant amount of learning interaction data is available [42]. In recent years, numerous research works have been proposed on deep learning KT models, we will introduce deep learning models from the following four subcategories:

1) DKT;
2) Memory-aware KT;
3) AKT;
4) GKT.

1) Deep KT: DKT is the pioneering approach that introduces deep learning to complete the KT task. DKT employs RNNs [17] to process the input sequence of learning interactions over time, maintaining a hidden state that implicitly contains information about the history of all past elements of the sequence. This hidden state evolves based on both the previous knowledge state and the present input learning interaction [18]. DKT provides a high-dimensional and continuous representation of the knowledge state, enabling it to more effectively model the complex learning process. Typically, RNNs' variants, the long short-term memory (LSTM) networks [43], are more frequently used in the implementation of DKT, which is further strengthened by considering forgetting.

Fig. 5 illustrates the process of DKT. In DKT, exercises are represented by their contained KCs. For datasets with different numbers of KCs, DKT applies two different methods to convert students' learning interactions $\boldsymbol{X} = \{(e_1, a_1), (e_2, a_2), \ldots, (e_t, a_t), \ldots, (e_N, a_N)\}$ into a sequence of fixed-length input vectors. More specifically, for datasets with a small number K of unique KCs, $\boldsymbol{x}_t \in \{0, 1\}^{2K}$ is set as a one-hot embedding, where $\boldsymbol{x}_t^k = 1$ if the answer a_t of the exercise with KC k was correct or $\boldsymbol{x}_t^{k+K} = 1$ if the answer was incorrect. For datasets with a large number of unique KCs, one-hot embedding is considered too sparse. Therefore, DKT assigns each input vector $\boldsymbol{x}_t$ to a corresponding random vector, and then uses the embedded learning sequence as the input of RNNs. A linear mapping and activation function is then applied to the output hidden states to obtain the knowledge state of students

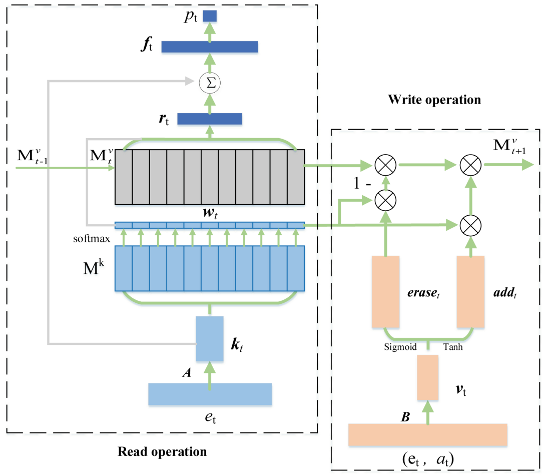

Fig. 6. Architecture of dynamic key.value memory network (DKVMN) [37].

$$\boldsymbol{h}_t = \tanh\left(\boldsymbol{W}_{hs}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h\right)$$
$$\boldsymbol{y}_t = \sigma\left(\boldsymbol{W}_{yh}\boldsymbol{h}_t + \boldsymbol{b}_y\right) \tag{7}$$

where tanh is the activation function, $\boldsymbol{W}_{hs}$ is the input weights, $\boldsymbol{W}_{hh}$ is the recurrent weights, $\boldsymbol{W}_{yh}$ is the readout weights, and $\boldsymbol{b}_h$ and $\boldsymbol{b}_y$ are the bias terms.

Despite demonstrating superior performance compared with Bayesian and logistic models, DKT has several inherent shortcomings. For instance, the lack of interpretability is a significant drawback. It is challenging to understand how the hidden states represent students' knowledge states, and the model cannot explicitly determine a student's knowledge mastery from the hidden state [42]. In addition, Yeung and Yeung [44] identified two unreasonable phenomena in DKT that contravene common sense. These are: 1) the inability to reconstruct observed input and 2) inconsistent predicted knowledge states across time steps. However, despite these shortcomings, DKT remains a promising KT model [45].

2) Memory-Aware KT: To enhance the interpretability of DKT, memory-aware KT introduces an external memory module, as proposed by Graves et al. [46]. This module is designed to store and update the corresponding knowledge mastery of the student. The most representative example is dynamic key–value memory networks (DKVMNs) for KT, as proposed by Zhang et al. [37]. DKVMN highlights students' specific knowledge states on various knowledge categories. It initializes a static matrix, referred to as a key matrix to store latent KCs and a dynamic matrix, called a value matrix to store and update the mastery of corresponding KCs through read and write operations over time.

As shown in Fig. 6, an embedding matrix is first defined to obtain the embedding vector k_t of the exercises. A correlation weight $\boldsymbol{w}_t$ is then obtained by taking the inner product between the exercise embedding k_t and the key vectors M^k, followed by the Softmax activation:

$$\boldsymbol{w}_t = \text{Softmax}\left(k_t M^k\right) \tag{8}$$

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

where the correlation weight $\boldsymbol{w}_t$ represents the correlation between the exercises and all latent KCs.

In the read operation, DKVMN predicts student performance based on the student's knowledge mastery. Specifically, DKVMN reads students' mastery of the exercise $\boldsymbol{r}_t$ with reference to the weighted sum of all memory vectors in the value matrix using the correlation weight. The read content and the input exercise embeddings are then concatenated together and passed to a fully connected layer to yield a summary vector $\boldsymbol{f}_t$, which contains both the student's knowledge mastery and the prior difficulty of the exercise. Furthermore, the student's performance can be predicted by applying another fully connected layer with a sigmoid activation function to the summary vector

$$\begin{aligned}\boldsymbol{r}_t &= \sum_{i=1}^{N} w_t(i) M_t^v(i)\\ \boldsymbol{f}_t &= \tanh\left(\boldsymbol{W}_f\left[\boldsymbol{r}_t, k_t\right] + \boldsymbol{b}_f\right)\\ p_t &= \sigma\left(\boldsymbol{W}_p \boldsymbol{f}_t + \boldsymbol{b}_p\right)\end{aligned} \tag{9}$$

where $\boldsymbol{W}_f$ and $\boldsymbol{W}_p$ are the weights, and $\boldsymbol{b}_f$ and $\boldsymbol{b}_p$ are bias terms.

In the write operation, after an exercise has been answered, DKVMN updates students' knowledge mastery (i.e., the value matrix) based on their performance. Specifically, the learning interaction (e_t, a_t) is first embedded with an embedding matrix $\boldsymbol{B}$ to obtain the student's knowledge growth $\boldsymbol{v}_t$. Then, DKVMN calculates an erase vector $\boldsymbol{erase}_t$ from $\boldsymbol{v}_t$ and decides to erase the previous memory with reference to both the erase vector and the correlation weight $\boldsymbol{w}_t$. Following erasure, the new memory vectors are updated by the new knowledge state and the add vector $\boldsymbol{add}_t$, which forms an erase-followed-by-add mechanism that allows forgetting and strengthening knowledge mastery in the learning process:

$$\begin{aligned}\boldsymbol{erase}_t &= \sigma\left(\boldsymbol{W}_e \boldsymbol{v}_t + \boldsymbol{b}_e\right)\\ \widetilde{M_t^v}(i) &= M_{t-1}^v(i)\left[1 - w_t(i)\boldsymbol{erase}_t\right]\\ \boldsymbol{add}_t &= \tanh\left(\boldsymbol{W}_d \boldsymbol{v}_t + \boldsymbol{b}_d\right)\\ M_t^v(i) &= \widetilde{M_t^v}(i) + w_t(i)\boldsymbol{add}_t\end{aligned} \tag{10}$$

where $\boldsymbol{W}_e$ and $\boldsymbol{W}_d$ are the weights, and $\boldsymbol{b}_e$ and $\boldsymbol{b}_d$ are bias terms.

Abdelrahman and Wang [47] pointed out that DKVMN failed to capture long-term dependencies in the learning process. Therefore, they propose a sequential KVMN (SKVMN) to combine the strengths of DKT's recurrent modeling capacity and DKVMN's memory capacity. In SKVMN, a modified LSTM called *Hop-LSTM* is used to hop across LSTM cells according to the relevance of the latent KCs, which directly captures the long-term dependencies. During the writing process, SKVMN allows for the calculation of the knowledge growth of a new exercise, taking into consideration the current knowledge state, thereby yielding more reasonable results.

3) Attentive KT: In the development of deep learning, the Transformer is initially proposed for neural machine translation [25], which abandons recurrence and solely relies on the self-attention mechanism to capture global dependencies within a sequence. The Transformer has been demonstrated to excel in feature extraction and dependency capture, while maintaining high computational efficiency. Some representative pretraining models based on the Transformer, such as BERT [48] and GPT [49], have obtained state-of-the-art results on various natural language processing tasks. Pandey and Karypis [20] proposed a self-attentive model for KT (SAKT), which directly apply the Transformer to capture long-term dependencies between students' learning interactions. Furthermore, Wang et al. [50] introduced an adaptive sparse self-attention network to generate missing features and simultaneously produce fine-grained predictions of student performance. Zhu et al. [51] employed a multihead ProbSparse self-attention mechanism to mitigate the time complexity and effectively capture the long-term dependencies in students' learning interactions.

However, the complexity of the KT task often limits the performance of the aforementioned simple Transformer applications. Choi et al. [26] introduced a novel approach named separated self-attentive neural KT (SAINT) to enhance self-attentive computation for KT adaptability. Specifically, SAINT employs an encoder–decoder structure, with the exercise and answer embeddings being separately encoded and decoded by self-attention layers. The separation of the input allows SAINT to stack self-attention layers multiple times, thus capturing complex relations in student interactions. Subsequently, Shin et al. [27] introduced the SAINT+ model, which integrates two temporal features into SAINT: namely, the time taken to answer each exercise and the interval time between consecutive learning interactions. Both SAINT and SAINT+ have outperformed the SAKT model on the student performance prediction task.

In addition, Ghosh et al. [21] observed that SAKT does not surpass DKT and DKVMN in their experiments. Unlike SAINT and SAINT+, they present a context-aware AKT model. This model integrates the self-attention mechanism with psychometric models, creating a more effective system. AKT is composed of four modules: Rasch model-based embeddings, exercise encoder, knowledge encoder, and knowledge retriever. Specifically, the embedding module employs the classic Rasch model in psychometrics [52] to construct embeddings for exercises and KCs

$$\boldsymbol{x}_t = \boldsymbol{c}_{c_t} + \mu_{e_t} \cdot \boldsymbol{d}_{c_t} \tag{11}$$

where $\boldsymbol{c}_{c_t} \in \mathbb{R}^{\boldsymbol{D}}$ is the embedding of the KC of this exercise, $\boldsymbol{d}_{c_t} \in \mathbb{R}^{\boldsymbol{D}}$ is a vector that summarizes the variation in exercises with the related KC, and $\mu_{e_t} \in \mathbb{R}^{\boldsymbol{D}}$ is a scalar difficulty parameter that controls the extent to which this exercise deviates from the related KC. The exercise–answer tuple (e_t, a_t) is similarly extended using the scalar difficulty parameter for each pair

$$\boldsymbol{y}_t = \boldsymbol{q}_{(c_t, a_t)} + \mu_{e_t} \cdot \boldsymbol{f}_{(c_t, a_t)} \tag{12}$$

where $\boldsymbol{q}_{(c_t,a_t)} \in \mathbb{R}^{\boldsymbol{D}}$ is the KC-answer embedding, and $\boldsymbol{f}_{(c_t,a_t)} \in \mathbb{R}^{\boldsymbol{D}}$ is the variation vector. Through the above embedding, exercises labeled as the same KCs are determined to be closely related while retaining important individual characteristics. Then, in the exercise encoder, the input is the exercise embeddings $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t\}$ and the output is a sequence of context-aware exercise embeddings $\{\widetilde{\boldsymbol{e}}_1, \ldots, \widetilde{\boldsymbol{e}}_t\}$. AKT designs a monotonic attention mechanism to accomplish the above process,

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

where the context-aware embedding of each exercise depends on both itself and the previous exercises, i.e., $\widetilde{\boldsymbol{e}}_t = f_{\text{enc}_1}(\boldsymbol{e}_1, \ldots, \boldsymbol{e}_t)$. Similarly, the knowledge encoder takes exercise–answer embeddings $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ as input and outputs a sequence of context-aware embeddings of the knowledge acquisitions $\{\widetilde{\boldsymbol{y}}_1, \ldots, \widetilde{\boldsymbol{y}}_t\}$ using the same monotonic attention mechanism; these are also determined by students' answers to both the current exercise and prior exercises, i.e., $\widetilde{\boldsymbol{y}}_t = f_{\text{enc}_1}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Finally, the knowledge retriever takes the context-aware exercise embedding $\widetilde{\boldsymbol{e}}_{1:t}$ and exercise–answer pair embeddings $\widetilde{\boldsymbol{y}}_{1:t}$ as input and outputs a retrieved knowledge state $\boldsymbol{h}_t$ for the current exercise. Since the student's current knowledge state depends on answering the related exercise, it is also context-aware in AKT. The novel monotonic attention mechanism proposed in AKT is based on the assumption that the learning process is temporal and students' knowledge will decay over time. Therefore, the scaled inner product attention mechanism utilized in the original Transformer is not suitable for the KT task. AKT uses exponential decay and a context-aware relative distance measure to compute the attention weights. Finally, AKT achieves outstanding performance in predicting students' future answers, as well as demonstrating interpretability due to the combination of the psychometric model.

It is important to note that Pu and Becker [53] have recently proposed that AKT models significantly benefit from students' continuous, repeated interactions on the same exercises throughout the learning process. In their experiments, the removal of these repeated interactions in the dataset led to a decline in AKT's performance, bringing it close to that of DKVMN. Moreover, according to the findings of Yin et al. [54], existing AKT models primarily trace patterns of a learner's learning activities, rather than their evolving knowledge states. Consequently, they developed the Transformer model to facilitate stable knowledge state estimation and tracing, rather than solely focusing on next performance prediction.

4) Graph-based KT: Graph neural networks, which are designed to handle complex graph-related data, have developed rapidly in recent years [55]. The graph represents a kind of data structure that models a set of objects (nodes) and their relationships (edges). From a data structure perspective, there is a naturally existing graph structure within the KCs. Therefore, incorporating the graph structure of the KCs as additional information should be beneficial to the KT task. Nakagawa et al. [19] presented GKT, which conceptualizes the potential graph structure of the KCs as a graph $G = (V, E)$, where nodes $V = \{v_1, v_2, \ldots, v_N\}$ represent the set of KCs and the edges $E \subseteq V \times V$ represent relationships of these KCs; moreover, $\boldsymbol{h}^t = \{\boldsymbol{h}^t_{i \in V}\}$ represents the student's temporal knowledge state after answering the exercise at time t. The architecture for GKT is presented in Fig. 7, which is composed of the following three parts:

1) aggregate;
2) update;
3) predict.

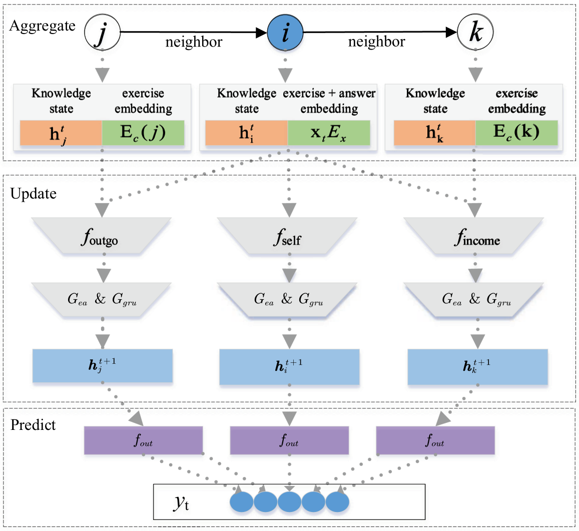

Fig. 7. Architecture of GKT [19].

In the aggregate module, GKT aggregates the temporal knowledge state and the embedding for the answered KC i and its neighboring KC j

$$\boldsymbol{h}'^t_k = \begin{cases} [\boldsymbol{h}^t_k, a^t \boldsymbol{E}_s], & (k = i) \\ [\boldsymbol{h}^t_k, \boldsymbol{E}_e(k)], & (k \neq i) \end{cases} \tag{13}$$

where a^t represents the exercises answered correctly or incorrectly at time step t, $\boldsymbol{E}_s$ is the embedding matrix for the learning interactions, $\boldsymbol{E}_e$ is the embedding matrix for the KC, and k represents the kth row of $\boldsymbol{E}_e$.

In the update module, GKT updates the temporal knowledge state based on the aggregated features and the knowledge graph structure, as follows:

$$\begin{aligned} \boldsymbol{m}^{t+1}_k &= \begin{cases} f_{\text{self}}\left(\boldsymbol{h}'^t_k\right) (k = i) \\ f_{\text{neighbor}}\left(\boldsymbol{h}'^t_i, \boldsymbol{h}'^t_k\right) (k \neq i) \end{cases} \\ \widetilde{\boldsymbol{m}}^{t+1}_k &= G_{\text{ea}}\left(\boldsymbol{m}^{t+1}_k\right) \\ \boldsymbol{h}^{t+1}_k &= G_{\text{gru}}\left(\widetilde{\boldsymbol{m}}^{t+1}_k, \boldsymbol{h}^t_k\right) \end{aligned} \tag{14}$$

where f_{self} is the multilayer perceptron, G_{ea} is the same erase-followed-by-add mechanism used in DKVMN, and G_{gru} is the gated recurrent unit (GRU) gate [56]. Moreover, f_{neighbor} defines the information propagation to neighboring nodes based on the knowledge graph structure.

In the predict module, GKT predicts the student's performance at the next time step according to the updated temporal knowledge state

$$y^t_k = \sigma(\boldsymbol{W}_k \boldsymbol{h}^{t+1}_k + \boldsymbol{b}_k) \tag{15}$$

where $\boldsymbol{W}_k$ is the weight parameter and $\boldsymbol{b}_k$ is the bias term.

In addition to modeling the graph structure in KCs by graph neural networks, Lu et al. [57] proposed to model the educational relation and topology in the concept map, which will be intended to act as mathematical constraints for the construction of the KT model. Recently, in the attempt to further explore knowledge structure, Tong et al. [58] proposed structure-based knowledge

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

tracing (SKT), which aims to capture the multiple relations in knowledge structure to model the influence propagation among concepts. SKT is mainly motivated by an education theory, *transfer of knowledge* [59], which claims that students' knowledge states on some relevant KCs will also be changed when they are practicing on a specific KC due to the potential knowledge structure among KCs. Therefore, a student's knowledge state is determined by not only the temporal effect from the exercise sequence, but also the spatial effect from the knowledge structure. To concurrently model the latent spatial effects, SKT presents the synchronization and partial propagation methods to characterize the undirected and directed relations between KCs, respectively. In this way, SKT measures influence propagation in the knowledge structure with both temporal and spatial relations. To get rid of dependence on knowledge structure, Long et al. [60] proposed the automatical GKT, which utilizes the automatical graph to measure students' knowledge states automatically without annotation manual annotations.

D. Summarization

It is crucial to emphasize that, despite deep learning models exhibiting superior performance compared with Bayesian models and logistic models, there remains a significant room for improvement in their interpretability and explainability. Due to the end-to-end learning strategy, deep learning models are notoriously difficult to interpret. The modeling process itself is also challenging to explain. Specifically, deep learning models are predominantly data driven and benefit a lot from large-scale student learning data. It is challenging to understand how they calculate a student's knowledge state with no theoretical guidance [33]. All we have are the results generated by these models. Therefore, any errors made by these models will lead students to doubt their reliability. The lack of explainability and interpretability has, thus, limited their further applicability.

To make the complex KT models interpretable, especially those deep learning models, researchers have attempted various methods. Lu et al. [61] presented a posthoc approach to reveal the interpretability of DKT. Specifically, they employed the layerwise relevance propagation technique [62] to interpret DKT by measuring the relevance between DKT's output and input. Preliminary experimental results suggest that this posthoc approach could be a promising method for explaining DKT. Besides, explainable AI (xAI) is proposed to make the black-box deep learning models more transparent, thereby promoting its applications [63]. Lu et al. [64] proposed to use the xAI technique to interpret the complex KT models based on deep learning. The interpreting results of DKT have been demonstrated to aid in enhancing the trust of students and teachers. Their findings suggested that it is promising to utilize xAI techniques to interpret the deep learning KT models, thereby assisting users in accepting and applying the suggestions provided by these models.

IV. Variants of KT Models

The aforementioned fundamental KT models are typically based on simplistic assumptions regarding the learning process. Specifically, these models predominantly employ learning interactions, such as exercises and responses to estimate students' knowledge states. However, the learning process is not solely represented by exercises and responses, but is influenced by various factors. In summary, the aforementioned fundamental KT models, while straightforward, may have reduced performance in real-world learning scenarios. Consequently, numerous variants have been proposed under more stringent assumptions, reflecting a more comprehensive learning process in real-world scenarios. Accordingly, we classify and review current variants of fundamental KT models into the following four categories:

1) modeling individualization before learning;
2) incorporating engagement during learning;
3) considering forgetting after learning;
4) utilizing side information across learning.

A. Modeling Individualization Before Learning

Everything and everyone possess unique characteristics. For instance, Liu et al. [65] explored several personalized factors of various tourists to recommend personalized travel packages. Similarly, the concept of individualization in the KT task implies that different students often exhibit different learning characteristics (such as varying learning rates or prior knowledge). Considering the student-specific variability in learning could potentially enhance the KT process, as suggested by Yudelson et al. [66]. In the subsequent sections, we will introduce various variant KT models that take into account individualization before learning.

The initial BKT paper has delved into the concept of individualization. Specifically, it uses all students' learning interactions on a specific KC to learn the individual parameter. Similarly, for a specific student, all his or her learning interactions are utilized to fit his or her individual learning parameters [9]. Consequently, BKT is able to ascertain different learning and performance parameters for various students and KCs. However, this approach only offers a marginal improvement compared with original BKT.

Subsequently, Pardos and Heffernan [67] proposed two simple variants of BKT that, respectively, individualize students' initial probability of mastery and the probability of transition from the unlearned state to the learned state. Specifically, a student node is added to individualize the initial probability of mastery for each student. The student node assigns each student with a personalized initial probability of mastery. A conditional probability table is designed to determine the value of the student node. Similarly, if changing the connection of the student node to the subsequent knowledge nodes, the transition probability parameter can also be individualized. In this case, the student node gives individualized transition parameters to each student. Moreover, rather than individualizing only one kind of parameter in BKT, some other variants of BKT opt to individualize all four BKT parameters simultaneously [66]. Lee and Brunskill [68] suggested that when applied in an intelligent tutoring system, the individualized BKT model can yield good improvements to student learning efficacy, reducing by about half the amount of questions required for 20% of students to achieve mastery.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

Another means of modeling individualization is clustering, which considers a wider range of students in different groups [69]. By clustering the students into various groups, we can train different KT models and make predictions on the test data. The number of clusters is, then, varied according to the student groups, and the predicting process is repeated iteratively. Finally, we can obtain a set of different predictions. Furthermore, there are two common methods used to combine these predictions [70]. 1) Uniform averaging, which simply averages the predictions and 2) weighted averaging, which combines the models by means of a weighted average. To realize clustering, K-means is a basic clustering algorithm that randomly initializes a set of cluster centroids, which are identified using Euclidean distance. An another popular clustering algorithm is spectral clustering, which represents the data as an undirected graph and analyzes the spectrum of the graph Laplacian obtained from the pairwise similarities of data points. Recently, some novel clustering algorithms have been proposed, including discrete nonnegative spectral clustering [71] and clustering uncertain data [72].

Minn et al. [73] proposed a model named DKT with dynamic student classification (DKT-DSC), which introduces individualization to DKT by exploiting the idea of clustering. According to students' previous performance, DKT-DSC assigns students with a similar learning ability to the same group. The knowledge states of students in different groups are, then, traced by different DKT models. Moreover, considering the dynamic property of the learning ability, each student's learning sequence is segmented into multiple time intervals. At the start of each time interval, DKT-DSC will reassess students' learning ability and reassign their groups. In DKT-DSC, the K-means clustering algorithm is utilized to split students with similar ability levels into the same group at each time interval. After learning the centroids of all K clusters, each student is assigned to the nearest cluster. Through dynamic student clustering, DKT-DSC offers an effective approach to realizing individualization in DKT.

Mao et al. [74] claimed that it is significant to consider both individual exercise representation and individual prior knowledge. They proposed a fine-grained knowledge tracing model, named FGKT. FGKT obtains the individual exercise representation through the acquisition of knowledge cells and exercise distinctions. Subsequently, it assesses the individual prior knowledge by evaluating the relevance between current and historical learning interactions. Finally, the above individual representations will be utilized as the input of LSTM in FGKT to evaluate students' evolving knowledge states. Zhao et al. [75] also noticed that the individualization of exercises is significant for measuring students' knowledge states. They proposed to consider multiple exercise factors, including the difficulty and the discrimination, to enhance the performance of DKT.

Shen et al. [76] proposed a convolutional knowledge tracing (CKT) model to implicitly measure student individualization. Specifically, CKT considers two factors that influence students' individualization: individualized learning rates and individualized prior knowledge. Individualized learning rates represent students' differing capacities to absorb knowledge. The sequence of student learning interactions can reflect different learning rates in the sense that students with high learning rates can rapidly master knowledge, while others need to spend more time trying and failing. Therefore, it is reasonable to assess the differences in learning rate by simultaneously processing several continuous learning interactions within a sliding window of convolutional neural networks [77]. Besides, individualized prior knowledge refers to students' prior knowledge, which can be assessed via their historical learning interactions.

B. Incorporating Engagement During Learning

Student engagement is defined as *"the quality of effort students themselves devote to educationally purposeful activities that contribute directly to desired outcomes"* [78]. This definition highlights a strong connection between student engagement and the learning process. Generally, higher engagement leads to enhanced knowledge gains. Consequently, considering student engagement in the learning process could potentially improve KT results [79]. In this section, we will present some variants that integrate student engagement into the KT models.

Student engagement is difficult to be directly measured. In practice, some online learning systems have made use of sensor data to measure student engagement. For example, inexpensive portable electroencephalography (EEG) devices can help to detect a variety of student mental states in learning, which can be seen as reflections of student engagement [80]. Xu et al. [81] proposed two methods that combine EEG-measured mental states to improve the performance of BKT. Concretely, the first one inserts a 1-D binary EEG measure into BKT, forming the EEG-BKT structure that extends BKT by adding a binary variable node E between the knowledge node and the answer node. The second one, i.e., EEG-BKT, utilizes logistic regression to combine an m-dimensional continuous variable E extracted from the raw EEG signal in BKT.

However, in most cases, it is difficult to collect sensor data on every student. Therefore, Schultz and Arroyo [82] proposed the knowledge and affect tracing (KAT) to model both knowledge and engagement in parallel. KAT is a sensorless model that does not rely on any sensor data. In this model, both knowledge and engagement are assumed to have direct influences on student performance. KAT considers three kinds of disengagement behaviors: quick guess (the student makes an attempt very quickly), bottom-out hint (all available hints are used), and many attempts (making more than three attempts at an exercise). These three behaviors are grouped as "gaming" behaviors in order to predict students' knowledge and engagement at each learning interaction. Rather than assuming equal influence of knowledge and engagement on students' knowledge state, one variation on the KAT model defines the connection between knowledge and engagement, and accordingly considers that students' knowledge states will influence their engagement. For example, students are more likely to disengage from knowledge they are not familiar with. Moreover, rather than explicitly modeling student engagement, Schultz and Arroyo [83] further proposed the knowledge tracing with behavior (KTB) model, which has only one latent knowledge node that acts as a combination of both knowledge and engagement. KTB assumes that both

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

engagement and performance are expressions of knowledge. The Bayesian estimation of the knowledge state needs to be inferred by both student engagement and performance.

Mongkhonvanit et al. [84] proposed to add five features in the process of watching videos on massive open online courses (MOOCs) to the input of DKT. These features reflect student engagement from various aspects, including playback speed, whether or not the video was paused, fast forwarded or rewound, and whether or not the video was completed. For example, if a student watches a video at a much faster playback speed, it is likely that he/she is impatient and absent minded. This model incorporates two further features: whether or not the exercise was submitted with an answer selected and whether or not the exercise was a part of an end-of-unit quiz, both of which are considered together. Experimental results indicate that DKT can achieve better performance through incorporating the above binarized engagement covariates.

C. *Considering Forgetting After Learning*

In real-world scenarios, while learning, forgetting is inevitable [85]. The *Ebbinghaus forgetting curve theory* indicates that students' knowledge proficiency will decline due to forgetting [86]. Recently, Huang et al. [87] proposed the concept of "knowledge proficiency tracing," a model that can dynamically capture the changes in students' proficiency levels on knowledge concepts over time. This model effectively tracks these changes in an interpretable manner. Therefore, the assumption that students' knowledge states will remain constant over time is untenable. However, fundamental KT models, such as the BKT, often overlook forgetting. In the following, we will introduce some variants of fundamental KT models that have attempted to consider forgetting after learning for more precise knowledge states.

Qiu et al. [88] discovered that BKT consistently overestimates the accuracy of students' answers when a day or more had elapsed since their previous responses. The underlying reason is that BKT assumes that student performance will remain the same regardless of how much time has passed. To consider how student performance declines with time, they proposed a BKT-forget model, which hypothesizes that students may forget information they have learned as days go by. In the BKT-forget model, a time node is added to specify which parameters should be affected by a new day and the new day node is fixed with a prior probability of 0.2. It also introduced parameters to represent the forgetting rate on a new day and denote the forgetting rate on the same day. However, although BKT-forget does consider the decline in student performance, it can only model forgetting that occurs over the time scale of days. To model the continuous decay of knowledge as time progresses, Nedungadi and Remya [89] incorporated forgetting into BKT based on the assumption that learned knowledge decays exponentially over time [90]. An exponential decay function is, thus, utilized to update the knowledge mastery level. They further assumed that the chance of forgetting will increase if a student does not practice the knowledge concepts within 30 days. Moreover, Khajah et al. [42] introduced an approach that counts the number of intervening trials and treats each as an independent opportunity for forgetting to occur.

Recall the PFA model in (5), in which the probability of students' mastery is estimated using a logistic function: $p(\theta) = \sigma(\beta + \mu s + \nu f)$. The original PFA model ignores the order of answers, in addition to the time between learning interactions. It is, therefore, difficult to directly incorporate time information into the original PFA model. Pelánek et al. [91] proposed PFA Elo/Extended (PFAE), a variant of the PFA model that combines PFA with some aspects of the Elo rating system [92]. The Elo rating system is originally devised for chess rating (estimating players' skills based on match results). In PFAE, θ is updated after each learning interaction

$$\theta := \begin{cases} \theta + \mu \cdot (1 - p(\theta)), & \text{if the answer was correct} \\ \theta + \nu \cdot p(\theta), & \text{if the answer was wrong.} \end{cases} \tag{16}$$

As the forgetting behavior of students is closely related to time, in order to consider forgetting, Pelánek [93] added a time effect function f to θ, i.e., using $p(\theta + f(t))$ instead of $p(\theta)$, where t is the time (in seconds) from the last learning interaction, and f is the time effect function.

To represent the complex forgetting behavior, the DKT-forget model [94] introduces forgetting into DKT, which considers the following three types of side information related to forgetting:

1) the repeated time gap that represents the interval time between the present interaction and the previous interaction with the same KC;
2) the sequence time gap that represents the interval time between the present interaction and the previous interaction;
3) past trial counts that represent the number of times a student has attempted on the exercise with the same KC.

All these three features are discretized at $\log_2$ scale. These side information are concatenated as additional information and represented as a multihot vector $\boldsymbol{c}_t$, which are integrated with the embedding vector $\boldsymbol{v}_t$ of the learning interaction, as follows:

$$\boldsymbol{v}_t^c = \theta^{\text{in}}(\boldsymbol{v}_t, \boldsymbol{c}_t) \tag{17}$$

where θ^{in} is the input integration function. The integrated input $\boldsymbol{v}_t^c$ and the previous knowledge state $\boldsymbol{h}_{t-1}$ are passed through the RNNs to update $\boldsymbol{h}_t$ in the same way as in (7). The additional information at the next time step $\boldsymbol{c}_{t+1}$ is also integrated with the updated $\boldsymbol{h}_t$

$$\boldsymbol{h}_t^c = \theta^{\text{out}}(\boldsymbol{h}_t, \boldsymbol{c}_{t+1}) \tag{18}$$

where θ^{out} is the output integration function.

Wang et al. [95] proposed a novel Hawkes KT model, which introduces the Hawkes process to adaptively model temporal cross-effects. The Hawkes process performs well at modeling sequential events localized in time, as it controls corresponding temporal trends by the intensity function. The intensity function in HawkesKT is designed to characterize the accumulative effects of previous learning interactions, along with their evolutions over time. In HawkesKT, the temporal cross-effects and the ways in which they evolve between historical learning interactions combine to form a dynamic learning process.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

D. Utilizing Side Information Across Learning

Most KT models primarily rely on exercises and student responses to evaluate students' knowledge states. These models have yielded impressive results and have been effectively implemented in online learning systems. Despite this, there are various other types of side information collected across the learning process that could be utilized to enhance these models. In this section, we will introduce several variants that aim to leverage this diverse side information across learning.

In terms of a student's first response time, a short initial response time could indicate either high proficiency or "gaming" behavior, while a long initial response time could indicate either careful thinking or lack of concentration. Since the connection between initial response time and knowledge state could be influenced by complex factors, Wang and Heffernan [96] proposed to discretize the continuous first response time into four categories (i.e., extremely short, short, long, and extremely long) to eliminate unnecessary information and simplify the latent complex possibilities. They, then, build a one-by-four parameter table, in which each column represents the category of the initial response time of the previous exercise, while the relevant values represent the probability of correct answers.

Regarding tutor intervention, Beck et al. [97] proposed the Bayesian evaluation and assessment model, which simultaneously assesses students' knowledge states and evaluates the lasting impact of tutor intervention. More specifically, it adds one observable binary intervention node to BKT: *True* means that the tutor intervention occurs in corresponding interactions while *False* indicates the opposite. The connection between the intervention node and knowledge node indicates the potential impact of the tutor intervention on students' knowledge states. The intervention node is linked to all four BKT parameters. As a result, there are a total of eight parameters to learn in order to incorporate tutor intervention. One possible way to reduce the number of parameters is choosing to link only the intervention node to the learning rate parameter [98]. Similarly, Lin and Chi [99] developed the intervention-BKT model, which incorporates two types of interventions into BKT and distinguishes their different effects: *elicit and tell*. The relations between the intervention and performance nodes represent the impact of teaching interventions on student performance, while the relations between the intervention and knowledge nodes represent the impact of teaching interventions on students' knowledge states. Therefore, at each learning interaction, while the present knowledge state is conditional on both the previous knowledge state and the current intervention, the student's performance depends on both the present knowledge state and the current intervention.

Rather than considering only one kind of side information, González-Brenes et al. [100] proposed a feature-aware student KT (FAST) model, which allows for the utilization of all kinds of side information. Traditional BKT uses conditional probability tables for the guessing, slipping, transition, and learning probabilities, meaning that the number of features involved in inference grows exponentially. Therefore, as the number of features increases, the time and space complexities of the model also grow exponentially. To deal with this large number of features, FAST uses logistic regression parameters rather than conditional probability tables. The number of features and complexities increase linearly rather than exponentially. For parameter learning, FAST uses the expectation maximization with features algorithm [101] and focuses on only emission features. The E step uses the current parameter estimates λ to infer the probability of the student having mastered the KC at each learning interaction. The parameters λ are now a function of the weight β and the feature vector $\boldsymbol{f}(t)$. $\boldsymbol{f}$ is the feature extraction function, and $\boldsymbol{f}(t)$ is the feature vector constructed from the observations at the relevant time step. The emission probability is represented with a logistic function

$$\lambda(\beta)^{y',k'} = \frac{1}{1+\exp(-\beta^T \cdot \boldsymbol{f}(t))} \tag{19}$$

where β is learned by training a weighted regularized logistic regression using a gradient-based search algorithm.

Zhang et al. [102] proposed an extension to DKT that explored the inclusion of additional features. Specifically, it incorporates an autoencoder network layer to convert the higher dimensional input data into smaller representative feature vectors, thereby reducing both the resource and time requirement for training. Students' response time, opportunity count, and first action are selected as incorporated side information, and all input features are converted into a fixed-length input vector. First, all input features are converted into categorical data and represented as a sparse vector by means of one-hot encoding. These encoded features are concatenated together to construct the higher dimensional input vector

$$\begin{aligned} C\left(e_t, a_t\right) &= e_t + (\max(e)+1)\, a_t \\ v_t &= O\left(C\left(e_t, a_t\right)\right) \oplus O\left(C\left(t_t, a_t\right)\right) \oplus O(t_t) \\ v_t' &= \tanh\left(W_v v_t + b_v\right) \end{aligned} \tag{20}$$

where C is the cross feature, O is the one-hot encoder format, v_t represents the resulting input vector of each learning interaction, e_t is the exercise, a_t refers to the answer, t_t is the response time, W_v is the weight parameter, and b_v is the bias term. Subsequently, an autoencoder is introduced to reduce the dimensionality without incurring the loss of too much important information. Finally, the feature vectors extracted by autoencoder will be the new input of DKT.

Liu et al. [103] presented another extension to DKT, namely, the exercise-aware KT (EKT), which utilized the potential value of exercises' text contents. Generally, the text content is of great significance for students to understand the exercises. For example, Huang et al. [104] used text materials to automatically predict their difficulties, Liu et al. [105] utilized the text content to find similar exercises. Yin et al. [106] further proposed a pretraining model called QuesNet for learning the unified representations of heterogeneous exercises. Therefore, instead of using one-hot encoding of exercises, EKT automatically learns the semantic representation of each exercise from its text contents. EKT first uses Word2vec [107] to pretrain the embedding vector for each word in the exercise. It, then, constructs a

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

bidirectional LSTM, which captures the word sequence from both forward and backward directions to learn the semantic word representation. The elementwise max-pooling operation is utilized to merge words' contextual representations into a global embedding. Finally, EKT can update the student's knowledge state with the aid of the semantic representation of each exercise.

To achieve more feasible integration of side information, Loh [108] presented a DKT with decision trees, which takes advantage of classification and regression trees (CART) to preprocess the heterogeneous input features [109]. Specifically, CART is utilized to automatically partition the feature space and outputs whether or not a student can answer an exercise correctly. The predicted response and the true response are encoded into a four-b binary code; for example, the code is 1010 if the predicted response and the true response are both correct. This binary code is, then, concatenated with the original one-hot encoding of the exercise as the new input of DKT to train the corresponding model.

Jung et al. [110] suggested that the student's language proficiency can serve as supplementary information to improve existing KT models. The student's language proficiency is extracted by Elo rating score and time window features. Then, the language proficiency information is demonstrated to be effective in promoting several KT models, including DKT, DKVMN, and SAKT. In addition, the problem of cold start in the KT task is alleviated with the assistance of language proficiency information. Liu et al. [111] explored to add side information to the original KT model by auxiliary learning tasks. They specifically introduced two tasks: 1) predicting the KCs of the question and 2) predicting the individualized prior knowledge. By training with these tasks, KT can enhance its understanding of the intrinsic relationships between questions and KCs, while explicitly capturing student-level variability.

When solving programming problems, we can record students' full code submissions, which can be employed to analyze their programming ability. Kasurinen and Nikula [112] collected student programming data and analyzed their personalized programming preferences. The study commenced with a statistical analysis of student errors, followed by an examination of students' programming structures derived from their code submissions, and finally used BKT to measure students' programming abilities. Wang et al. [113] transformed students' code submissions into embedded vectors, and applied them in DKT to model students' fine-graded programming knowledge states. Zhu et al. [114] noticed that a single programming problem generally involves in multiple KCs, thereby they proposed to learn useful information about the programming problem's multiple requirements from students' code submissions.

V. Applications

Although KT is an emerging research area, it has already been applied in a wide variety of scenarios. In the following, we will first survey the applications of KT models in two typical educational scenarios: learning resources recommendation and adaptive learning. Then, we will discuss broader applications of KT beyond student learning.

A. Learning Resources Recommendation

Traditionally, learning resources for each student are selected in one of two ways. The first one requires teachers to manually select suitable resources that match students' knowledge levels. However, this approach requires substantial time and effort, and different teachers may have different preferences. The second one allows students themselves to freely choose resources to learn. However, this may result in students choosing too easy or too difficult materials that will not benefit their learning [115], leading to low learning efficiency. In recent years, the prevalence of intelligent tutoring systems and the development of KT methods have made it possible to automatically recommend appropriate exercises to each student based on artificially designed algorithms.

Exercises are the most common learning resources in learning. Given the inferred knowledge states, one common strategy is selecting the next exercise that will best advance students' knowledge acquisition. Desmarais and Baker [115] proposed two extensions of the original BKT model, which, respectively, considered exercises' difficulties and students' multiple-attempt behaviors. These two extensions are integrated into a BKT-sequence algorithm to recommend exercises to students based on their knowledge states. Specifically, BKT-sequence first determines the predicted range of scores for each exercise. It then computes an expected score for each exercise that the student should get to achieve mastery, which is dependent on their current knowledge state (for instance, a lower knowledge state will result in higher expected scores). Finally, the algorithm returns the exercise with a predicted score that is closest to that of the expected score. Therefore, as the knowledge state of a particular KC grows, more difficult exercises will be recommended, as harder exercises are associated with a lower predictive score. Experimental results have shown that students using the BKT-sequence algorithm were able to solve more difficult exercises, obtained higher performance and spent more time in the system than students who used the traditional approach. Moreover, students also expressed that the BKT-sequence algorithm was more efficient. Wan et al. [116] expanded the DKVMN model [37] to include the exercise's type and difficulty. This model is then used to assess students' knowledge state and subsequently recommends personalized exercises for each student in small private online courses. They conducted a randomized controlled trial to show that the proposed personalized exercise recommendation could enhance students' learning efficiency.

In addition to exercises, there are also some other types of multimodal learning resources, such as videos and figures. Machardy [117] utilized an adaptation of BKT to improve student performance prediction by incorporating video observation. Experimental verification demonstrates the impact of both using and eschewing video data, as well as the learning rate associated with a particular video. In this way, he further developed a method to help people evaluate the quality of video resources. Concretely, they proposed the template 1 video model to incorporate video observations into BKT, which adds video activity as additional independent observation nodes to the BKT model. This model accordingly considers the probability that

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

a given video resource will impart knowledge to a student. Moreover, the transition probability in BKT is conditional only on the presence of either a video or an exercise. Thus, the quality of the video can be determined by its promotion of learning, and this model can be leveraged as a tool to aid in evaluating and recommending video resources.

When recommending learning resources, the primary aim of existing solutions is to choose a simple strategy for assigning nonmastered exercises to students. While reasonable, it is also too broad to advance learning effectively. Huang et al. [118] accordingly proposed three more beneficial and specific objectives: *review and explore*, *smoothness of difficulty level*, and *student engagement*. In more detail, *review and explore* considers both enhancing students' nonmastered concepts with timely reviews and reserving certain opportunities to explore new knowledge, *smoothness of difficulty level* indicates that the difficulty levels of several continuous exercises should vary within a small range as students gradually learn new knowledge, and finally, *student engagement* considers that to promote students' enthusiasm during learning, the recommended exercises should be in line with their preferences. In order to support online intelligent education with the above three domain-specific objectives, they developed a more reasonable multi-objective deep reinforcement learning (DRE) framework. DRE presented three corresponding novel reward functions to capture and quantify the effects of the above three objectives. This DRE framework is a unified platform designed to optimize multiple learning objectives, where more reasonable objectives also can be incorporated if necessary. Experimental results show that DRE can effectively learn from the students' learning records to optimize multiple objectives and adaptively recommend suitable exercises.

B. Adaptive Learning

Adaptive learning, unlike learning resource recommendations, goes beyond the mere provision of resources. It not only concentrates on the selection of appropriate learning materials but also designs effective learning strategies and dynamic learning pathways. These are structured based on both the learning rules and students' evolving knowledge states. Specifically, adaptive learning broadly refers to *"a learning process in which the content taught, or the way such content is presented, changes or "adapts" based on individual student responses, and which dynamically adjusts the level or types of instruction based on individual student abilities or preferences"* [119].

The first few attempt made to apply KT to adaptive learning was the ACT Programming Tutor [9], where students were asked to write short programs and BKT was utilized to estimate their evolving knowledge state. This tutor can present an individualized sequence of exercises to each student based on their estimated knowledge states until the student has "mastered" each rule.

In recent years, MOOCs have become an emerging modality of learning, particularly in higher education. Pardos et al. [11] adapt BKT on the edX platform. The research object was a 14-week online course that included weekly video lectures and corresponding lecture problems. BKT was applied to enhance students' learning on this course. In order to better adapt BKT to the learning platform, the original BKT was modified in several respects. First, due to the lack of labeled KCs, the problems would be directly seen as the KCs, while the questions would be seen as the exercises belonging to the KC. Second, in order to capture the varying degrees of students' knowledge acquisition at each attempt, the modified model assigned different guess and slip parameters to different attempt counts. Finally, to deal with the problem of multiple pathways in the system, which reflected that the impacts on learning may come from various resources, they framed the influence of resources on learning as a credit/blame inference problem.

Generally, students' cognitive structures include both students' knowledge level and the knowledge structure of learning items (e.g., *one-digit addition* is the prerequisite knowledge of *two-digit addition*). Therefore, adaptive learning should maintain consistency with both students' knowledge level and the latent knowledge structure. Nevertheless, existing methods for adaptive learning often focus separately on either the knowledge levels of students (i.e., with the help of specific KT models) or the knowledge structure of learning items. To fully exploit the cognitive structure for adaptive learning, Liu et al. [13] proposed a Cognitive Structure Enhanced framework for Adaptive Learning (CSEAL). CSEAL conceptualized adaptive learning as a Markov decision process. It first utilized DKT to trace the evolving knowledge states of students at each learning step. Subsequently, the authors designed a navigation algorithm based on the knowledge structure to ensure that the learning paths in adaptive learning were logical and reasonable, which also reduced the search space in the decision process. Finally, CSEAL utilized the actor–critic algorithm to dynamically determine what should be learned next. In this way, CSEAL can sequentially identify the most suitable learning resources for different students.

C. Broader Applications

The above two types of applications are most commonly used for KT in student learning. In addition, the KT methods can be expanded to be utilized in any systems that necessitate continuous evaluation of user capabilities or states. We will introduce some broader applications of KT in this section.

In gaming systems, the paradigm of tracing students' knowledge state can also work for player modeling. Here, player modeling, which is the study of computational models of players in games, aims to capture human players' characteristics and cognitive features [120]. For instance, Fisch et al. [121] revealed that children engage in cycles of increasingly sophisticated mathematical thinking over the course of playing an online game. Kantharaju et al. [122] presented an approach to trace player knowledge in a parallel programming educational game, which is capable of measuring the current players' real-time state across the different skills required to play an educational game based only on in-game player activities. Long and Aleven [123] conducted a classroom experiment comparing a commercial game for equation solving, i.e., *DragonBox*, with a research-based intelligent tutoring system, i.e., *Lynnette*. The

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

TABLE II
BASIC INFORMATION AND STATISTICS OF EXISTING DATASETS AVAILABLE FOR EVALUATING KT MODELS

Datasets	Subjects	Learning Stages	Side Information	Sources		Statistics			
				Online platforms	Educational challenges	# of Students	# of Exercises	# of KCs	# of Learning records
ASSISTments2009 [129]	Mathematics	Middle school	Yes	✓		4163	17 751	123	346 860
ASSISTments2012 [130]	Mathematics	Middle school	Yes	✓		46 674	179 999	265	6 123 270
ASSISTments2015	Mathematics	Middle school	No	✓		19 917	/	100	708 631
ASSISTments2017	Mathematics	From middle school to college	Yes		✓	1709	3162	102	942 816
Junyi [131]	Mathematics	From primary to high school	Yes	✓		247 606	722	41	25 925 922
Eedi2020 [132]	Mathematics	From primary to high school	Yes		✓	118 971	27 613	388	15 867 850
Statics2011 [133]	Engineering	University	Yes	✓		335	1224	80	361 092
EdNet-KT1 [134]	English	/	Yes	✓		784 309	13 169	188	95 293 926
EdNet-KT2 [134]	English	/	Yes	✓		297 444	13 169	188	56 360 602
EdNet-KT3 [134]	English	/	Yes	✓		297 915	13 169	293	89 270 654
EdNet-KT4 [134]	English	/	Yes	✓		297 915	13 169	293	131 441 538
CodeWorkout [135]	Programming	University	Yes		✓	819	50	50	>130 000

results indicated that students who used *DragonBox* enjoyed the experience more, while students who used *Lynnette* performed significantly better on the test. Therefore, it is possible to enable students to learn effectively and happily by designing suitable educational games on the online learning platform.

In crowdsourcing, unlabeled data or specific tasks are assigned to various crowd annotators. Understanding the dynamic capabilities of these annotators is crucial in ensuring the reliability of their annotations and promoting the annotation efficiency. Wang et al. [124] developed a framework called KT4Crowd, which utilized KT methods to predict the performance of annotators, which surpassed traditional rating systems. Besides, Abdi et al. [125] observed that students' participation in crowdsourcing tasks can enhance their learning. KT methods can also better comprehend students' knowledge states, aided by the students' annotated items.

Crowston et al. [126] developed an online citizen science project that employs machine learning techniques to improve the training of new volunteers using authentic tasks featuring uncertain outcomes such as image classification. Specifically, they employ the BKT model to monitor the knowledge states of volunteers, enabling them to more efficiently complete assigned tasks and contribute meaningfully to the project.

Zhao et al. [127] developed an automated exercise collection approach for teachers, employing the KT model and reinforcement learning. Specifically, the exercise collection need to be well designed to align with students' abilities. This study first leverages the KT model to forecast students' performance on unseen exercise candidates. Subsequently, the exercise selector is designed based on the KT model's predictions, ensuring that the exercise collection is both approximate and optimized. Similarly, Shang et al. [128] designed a reinforcement learning guided method for exam paper generation, where the DKT model is utilized to measure examinees' knowledge states.

VI. Datasets and Baselines

After introducing the above KT models and variants, to better help researchers and practitioners who want to further conduct related work and promote the application of KT, we have open sourced two algorithm libraries, i.e., EduData for downloading and preprocessing most existing KT-related datasets, and EduKTM that includes extensible and unified implementations of existing popular KT models. In the following, we will give detailed introduction of these two algorithm libraries.

A. *Datasets*

As we have mentioned, KT emerges from the development of online education, where a large number of students' learning data are collected for analyzing their learning behaviors and knowledge states. In this section, we mainly introduce existing public datasets available for evaluating KT models. Table II lists all datasets, as well as their basic information and statistics. In our released EduData, we provide the service of downloading and preprocessing all these datasets, which is convenient to help beginners analyze and utilize them quickly. In summary, these datasets are collected in different learning scenarios, so that they exhibit an extremely distinct difference in data scale, subject, and so on, indicating complex applications in practical for KT models.

1) ASSISTments Datasets: ASSISTments [129], created in 2004, is an online tutoring system in the United States, which provides students with both assessment information and tutoring assistance. While working on ASSISTments, students will be provided with instructional assistance to help them solve the problem in several substeps when they give wrong answers. After obtaining correct answers, they will be given a new one. Meanwhile, the system will learn about the students' knowledge states and predict how they will do in future tests. Up to now, the organizers have released four publicly available datasets from ASSISTments, which are, respectively, ASSISTments2009, ASSISTments2012,[3] ASSISTments2015,[4]

[3] [Online]. Available: https://sites.google.com/site/assistmentsdata/datasets/2012-13-school-data-with-affect

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

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

and ASSISTments2017.[5] The ASSISTments datasets have built profound impacts in the research community of KT, which can be seen in many related papers. Most of these datasets were collected from mathematics in middle school, the details of them are listed as follows.

1) *ASSISTments2009:*The full name of ASSISTments2009 is the ASSISTments 2009–2010 skill builder dataset [129], which was collected form ASSISTments' skill builder problem sets during the school year from 2009 to 2010. Students were asked to work on exercises with similar knowledge concepts until they can answer correctly for three or more times in a row. This dataset contains many valuable side information, such as *attempt count* that represents the number of attempts of the student, *ms first response* that represents the time in the milliseconds for the student's first response, and *opportunity* that represents the number of opportunities the student has to practice. It worth noting that the original version of ASSISTments2009 has the following three serious problems that lead to unreliable experimental results [45]:

a) a large number of duplicated records;
b) treating scaffolding problems as the same as main problems;
c) repeated response sequences with different KCs.

The latest version of ASSISTments2009 has fixed these problems.

2) *ASSISTments2012:* ASSISTments2012 is collected from the ASSISTments system, in the school year from 2012 to 2013 with affect predictions. Compared with ASSISTments2009, ASSISTments2012 has much more students, exercises, as well as learning records. However, many learning records have missed the related KCs in this dataset. After filtering these records, there are only 29 018 students, 53 091 exercises, and 2 711 813 learning records. Besides, this dataset contains more side information, such as *start time*, *end time*, *problem type*, and *average confidence*. It worth noting that you can access to the text of the exercises in ASSISTments2012 for more fine-grained research. Specifically, you can email nth@wpi.edu and cc td@wpi.edu, explaining your purpose and promising not to share it with anyone else.

3) *ASSISTments2015:* ASSISTments2015 covers the 2015 school years' student response records on ASSISTments. This dataset only contains student learning records on 100 KCs, there is no information about the exercise, as well as any other side information.

4) *ASSISTments2017:* ASSISTments2017 is the dataset used in the 2017 ASSISTments Longitudinal Data Mining Competition. This dataset collected data over a decade, which tracked students' interactions at the ASSISTments learning platform from middle school study in 2004–2007 to their high school course-taking, until they graduated from the college. Therefore, the average learning record of students in ASSISTments2017 is much longer than other dataset, which is beyond the length of 1000. ASSISTments2017 also has rich side information, including but not limited to *AveKnow* that indicates students' average knowledge level based on BKT, *timeTaken* that represents the time spent on the current exercise, and *frIsHelpRequest* that represents whether the first response is a help request.

2) Junyi Dataset: The Junyi dataset [131] contains the problem log and exercise-related information on the Junyi Academy,[6] a Chinese e-learning website established in 2012 on the basis of the open-source code released by Khan Academy. In contrast to the ASSISTments datasets, Junyi dataset has less exercises and KCs, but includes an exercise hierarchy labeled by experts, the annotations of exercise relationship are also available. Therefore, many research works that focused on the knowledge structure in KT had utilized this dataset [58]. Junyi dataset provides the prerequisite exercise of a specific exercise in the knowledge map, the topic and area of each exercise, as well as the coordinate position of the knowledge map.

3) Eedi2020 Dataset: The Eedi2020 dataset [132] is also released in an academic challenge, i.e., NeurIPS 2020 Education Challenge.[7] This dataset contains students' answers to mathematics questions from Eedi, an online educational platform where millions of students interact daily around the globe from school year 2018 to 2020. All exercises are multiple-choice problems with four possible answer choices, exactly one of which is correct. In Table II, we give the statistics based on the training data in this competition, the total number of learning records in the full dataset exceeds 17 million. It is worth noting that Eedi2020 gives students' exact answer choice so that we can also predict students options [136]. Moreover, for the students, Eedi2020 records lots of valuable context information, including the *Gender*, *DateOfBirth*, and *PremiumPupil*. For the learning records, Eedi2020 also presents their *Confidence*, *GroupId*, *QuizId*, and *SchemeOfWorkId*.

4) Statics2011 Dataset: Different from the above datasets that focus on mathematics exercises, the Statics2011 dataset [133] is obtained from a college-level engineering statics course via an online educational system developed by Carnegie Mellon University.[8] The problems in college engineering course are quite complex, often comprising numerous independent substeps. Consequently, we treat each subproblem as an exercise and calculate the number of exercises and KCs within this dataset.

5) EdNet Dataset: The EdNet dataset [134] is related to the English subject, which is consisted of students' learning records in the multiplatform AI tutoring system Santa in South Korea.[9] EdNet collected learning data of students over two years for their preparation of the e-Test of English for International Communication, listening and reading test. EdNet is now the largest public dataset in KT field with a total of 131 441 538 learning records from 784 309 students. Besides, it contains various features of

[5] [Online]. Available: https://sites.google.com/view/assistmentsdatamining/dataset

[6] [Online]. Available: http://www.junyiacademy.org/

[7] [Online]. Available: https://eedi.com/projects/neurips-education-challenge

[8] [Online]. Available: https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=507

[9] [Online]. Available: https://github.com/riiid/ednet

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

students' learning actions, such as the specific learning material they have interacted and how much time they have spent for answering a given exercise. There are four different versions of EdNet, respectively, named EdNet-KT1, EdNet-KT2, EdNet-KT3, and EdNet-KT4 with different extents. We note that the students in this dataset may be in different learning states. Therefore, we do not give the learning state information in Table II.

1) *EdNet-KT1:* EdNet-KT1 contains students' basic exercise-answering logs. This dataset has 784 309 students, 13 169 exercises, 188 KCs, and a total of 95 293 926 learning records. Exercises in EdNet-KT1 are organized by bundles, i.e., a collection of exercises sharing a common passage, picture, or listening material. Therefore, exercises come up in bundles and students have to answer all contained exercises when a bundle is given.
2) *EdNet-KT2:* EdNet-KT2 recorded students' action sequences, which indicated their full learning behaviors. For example, a student who is not confident about the answer may alternately select among several answer choices before submitting. Such learning behaviors can reflect more fine-grained knowledge state of students. EdNet-KT2 contains three kinds of actions: *enter* when student first receives and views a bundle, *respond* when the student selects an answer choice to the exercise, and *submit* when the student submits his final answers to the given bundle. It is worth noting that EdNet-KT2 is a subset of EdNet-KT1.
3) *EdNet-KT3:* On the basis of EdNet-KT2, EdNet-KT3 collected more students' learning activities, such as reading explanations or watching lectures. These learning activities have potential impacts on students' knowledge state so that they are valuable to be analyzed.
4) *EdNet-KT4:* In EdNet-KT4, the very fine details of actions were provided. In particular, the following types of actions are added to EdNet-KT3: erase choice, undo erase choice, play audio, pause audio, play video, pause video, pay, refund, and enroll coupon.

6) CodeWorkout Dataset: The CodeWorkout dataset [135] is utilized in the 2nd Computer Science Educational Data Mining Challenge.[10] This dataset is collected from a CS1 course in the Spring and Fall 2019 semesters at a public university in the United States. It contains the code submissions from students for 50 coding problems, each requiring 10–26 lines of code. In this dataset, each exercise has a unique KC, thereby the number of KCs is equal to the number of exercises. In total, there are 329 and 490 students in the Spring and Fall semesters, respectively, who completed the course. Each dataset contains more than 65 000 code submissions, the scores of the submissions (percentage of unit tests passes) are also available, as well as the compiler message if the compilation is not successful. The final grades of students are also provided for this dataset.

B. Baselines

The implementations of existing KT methods are not standardized, which may use different program languages (e.g., Python, Lua) and different deep learning frameworks (e.g., TensorFlow, torch). Furthermore, some works did not well organize the codes systematically (e.g., the missing of running environments and dependencies), which bring difficulties in reproducing the models. To this end, we put forward the algorithm library for KT baselines, named EduKTM, which now has contained the concurrent popular works. EduKTM will always be under development for including the latest KT models, more algorithms and features are going to be added. Besides, we provide detailed guidelines for everyone who is interested in contributing to EduKTM.

It is worth nothing that we do not provide the performance evaluation of these baselines using the aforementioned benchmark datasets. The reasons are twofold. 1) For each baseline method, their experimental settings are quite different as they were designed to handle various learning scenarios. No single KT model can always be the best one under various learning contexts. It is not fair to compare all baselines under a fixed setting. 2) Existing evaluation standards primarily concentrate on the performance of student performance prediction task, which cannot directly reflect the effectiveness of various KT models in practical applications. Therefore, we decide to open source EduData and EduKTM. Researchers and practitioners can freely compare and select appropriate KT models based on their specific requirements in various application scenarios.

VII. Future Research Directions

This survey has comprehensively reviewed the abundant current developments in the field of KT, including fundamental KT models, their variations, and typical applications. Despite this, as KT is a relatively new but promising research area, there are still a substantial number of problems that require urgent resolution. In this section, we discuss several potential future research directions.

A. KT With Interpretability

The performance of KT models is now evaluated indirectly through the student performance prediction task. The higher the precision of students' responses on future exercises, the better the KT model's performance. Nonetheless, interpretability plays a significant role in education, as students often express more concern about the "why" than the "what" of a learning decision [137]. Enhancing the interpretability of KT models is, therefore, crucial. Some educational theories, such as the Rasch model used in AKT [21] and the transfer of knowledge used in SKT [58], could be considered for this purpose. Minn et al. [138] noticed the significance of interpretable KT, particularly for deep learning models, as they have numerous parameters that are challenging to explain meaningfully. Thus, they [138] introduced a straightforward and interpretable KT approach, based on the causal relationships within the latent features extracted from students' behavioral data. Zhu et al. [29] attempted to introduce causal inference for explanatory analysis on KT, and they achieved more stable and explainable KT based on the analysis results. It is imperative to further refine existing KT models or to explore additional methods for interpretable KT

[10] [Online]. Available: https://sites.google.com/ncsu.edu/csedm-dc-2021/home

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

research works. This will lead to the production of more accurate and interpretable evaluations of students' knowledge states.

B. KT With Sparse Learning Interactions

The acquisition of high-quality KT models necessarily requires a substantial amount of data to ensure training stability. However, online learning systems often achieve less learning interactions in practical educational scenarios, thereby leading to the data sparsity problem. To address this issue, Yang et al. [139] utilized graph convolutional network to include exercise–KC correlations. Huang et al. [140] turned to improve the attention mechanism. More recently, researchers have employed contrastive learning to alleviate the data sparsity problem in KT [141], [142]. For example, Lee et al. [143] presented a contrastive learning framework to enhance KT, which measures the semantical similarity between various learning interactions to effectively learn their representations. They further designed data augmentation methods to enhance the semantics of students' learning interactions. However, while the aforementioned methods alleviate the data sparsity problem in various aspects, there remains a need for further improvement in addressing this issue comprehensively.

C. KT With Subjective Exercises

Most existing KT research works are capable of dealing with objective exercises, assuming that student responses are binary. However, they overlook students' open-ended answers, often presented in natural language, on subjective exercises. As per the work by Liu et al. [144], an open-ended KT approach for computer science education was explored. This method aimed to predict students' precise solutions to programming questions, utilizing language models for assistance. Recent advancements in large language models (LLM) offer promising potential to enhance the ability of KT models to comprehend the critical information regarding students' knowledge states embedded within their open-ended answers on subjective exercises [145].

D. KT With Students' Feedback

Learning records are passive representations of students' knowledge states. In contrast, students' feedback provides us with their proactive understanding of their knowledge states, thereby offering direct and authentic indicators of their learning situation. However, there are few KT models that take advantage of training data related to students' feedback, even though it can play an important role in fixing the KT results [146]. Wang et al. [147] noted that feedback plays a positive role in learning, which may promote transfer and retention in learning from worked-out examples. Therefore, incorporating students' feedback presents a meaningful avenue that could lead to significant improvements.

E. KT for General User Modeling

Generally, user modeling refers to tools for characterizing users' behaviors (e.g., frequent locations), personal information (e.g., age, gender, and occupation), and latent features (e.g., interests and abilities), which facilitate the provision of targeted services for different users [148]. As a type of latent feature modeling, KT diagnoses the proficiency of users (not only individuals, but also groups of individuals, such as user teams and companies) on specific skills/concepts. Thus, in addition to education, KT can be developed and applied across a wide range of domains for user modeling, including games, sports, and recruitment.

F. KT With LLM

In recent years, the advancement of LLM, notably ChatGPT, has led to significant impacts and garnered considerable research interest worldwide [149]. The use of LLM in education is promising to revolutionize the current learning pattern [145]. The current study mainly focuses on generating learning materials, improving student–system interaction, and explaining educational contents. However, it remains unclear how LLM can assist in understanding students' knowledge states. Given the powerful capabilities of LLM, it has the potential to enhance the generalization and interpretability of existing KT methods. Specifically, LLM can analyze the content of students' responses and evaluate their quality, which directly reflects students' knowledge states. Besides, LLM can play the role of the instructor, answering questions for students and identifying their strengths and weaknesses. Furthermore, LLM itself can serve as a KT model, which can output reasonable results about students' knowledge states given their previous learning interactions. Simultaneously, it is imperative to safeguard the privacy and security of student data when utilizing LLM [150].

VIII. Conclusion

In this survey, we conducted a thorough review of KT. Specifically, we initially conducted a comprehensive review of existing fundamental KT models. Considering the complexity of online learning systems and the significant importance of KT research in practical scenarios, we investigated a broad range of variant KT research. Subsequently, we summarized some typical applications of KT in common educational scenarios. Furthermore, we released two algorithm libraries for KT-related datasets and baselines, thereby facilitating researchers and practitioners in selecting appropriate KT models based on their specific requirements. Finally, we outlined some potential future directions.

We hope that this comprehensive survey of KT will assist readers in understanding the problem of modeling students' dynamic knowledge states. It serves as a fundamental framework for both researchers and practitioners in future studies, fostering the development of KT. The development of KT will directly benefit millions of students, with its impact potentially extending to a broader audience as online learning continues to evolve. In this context, KT will play an increasingly important role in enabling individuals to adapt to the ever-changing society [151]. Furthermore, the development of more refined KT methods tailored to students in various subjects and age groups will enhance their ability to comprehend students' individual knowledge states.

However, we recognize some limitations of the current survey. First, as new KT methods continue to emerge, although we have conducted a thorough investigation, there may be some representative works that we have overlooked. If necessary, we

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

will incorporate these into our proposed framework in the future. Second, the complexity of KT methods warrants attention, as it continues to grow, particularly in those based on deep learning. The complexity of the model may enhance its accuracy, but it could also compromise its applicability, as users will question the reliability of complex models. Despite the fact that some studies have attempted to reveal the interpretability of complex KT models, for instance, by utilizing the xAI technique, there remains a significant distance to traverse before achieving a completely transparent deep learning KT model.

We posit that the application of KT methods in online education presents a promising avenue for research. It significantly enhances the learning experience for students while simultaneously alleviating the burden on teachers. Despite the numerous challenges and obstacles, researchers are to be encouraged to overcome them and ensure a reliable and equitable access to students' evolving knowledge state in learning.

References

[1] G. Mckenna, "Theory and practice of online learning," *Int. J. Inf. Manage.*, vol. 29, no. 6, pp. 503–504, 2009.

[2] A. Sun and X. Chen, "Online education and its effective practice: A research review," *J. Inf. Technol. Educ. Res.*, vol. 15, no. 2016, pp. 157–190, 2015.

[3] M. Wang, M. Wang, Y. Cui, and H. Zhang, "Art teachers' attitudes toward online learning: An empirical study using self determination theory," *Front. Psychol.*, vol. 12, 2021, Art. no. 627095.

[4] K. Van Lehn, "The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems," *Educ. Psychol.*, vol. 46, no. 4, pp. 197–221, 2011.

[5] A. Cully and Y. Demiris, "Online knowledge level tracking with data-driven student models and collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 2000–2013, Oct. 2020.

[6] M. Guo, M. Wang, and B. Li, "A case study of mobile blended learning in teaching English as a foreign language," in *Proc. Intell. Environ. Workshops*, 2019, pp. 269–279.

[7] L. Jiang et al., "Reinforced explainable knowledge concept recommendation in MOOCs," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 3, pp. 1–20, 2023.

[8] L. Jiang, Y. Wang, S. Xie, J. Wu, M. Yin, and J. Wang, "Which courses to choose? Recommending courses to groups of students in online tutoring platforms," *Appl. Intell.*, vol. 53, no. 10, pp. 11727–11736, 2023.

[9] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modelling User-Adapted Interact.*, vol. 4, no. 4, pp. 253–278, 1995.

[10] C. E. Dowling and C. Hockemeyer, "Automata for the assessment of knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 3, pp. 451–461, May/Jun. 2001.

[11] Z. A. Pardos, Y. Bergner, D. T. Seaton, and D. E. Pritchard, "Adapting Bayesian knowledge tracing to a massive open online course in EDX," *Educ. Data Mining*, vol. 13, pp. 137–144, 2013.

[12] J. Cowan, "A learner centered approach to online education," *Brit. J. Educ. Technol.*, vol. 44, no. 6, pp. E221–E222, 2013.

[13] Q. Liu et al., "Exploiting cognitive structure for adaptive learning," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 627–635.

[14] J. H. Block and R. B. Burns, "Mastery learning," *Rev. Res. Educ.*, vol. 4, no. 1, pp. 3–49, 1976.

[15] H. Cen, K. Koedinger, and B. Junker, "Learning factors analysis–A general method for cognitive model evaluation and improvement," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2006, pp. 164–175.

[16] P. I. Pavlik, H. Cen, and K. R. Koedinger, *Performance Factors Analysis–A New Alternative to Knowledge Tracing*. Amsterdam, The Netherlands: IOS Press, 2009, pp. 531–538.

[17] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989.

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

[19] 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.

[20] S. Pandey and G. Karypis, "A self-attentive model for knowledge tracing," in *Proc. 12th Int. Conf. Educ. Data Mining*, 2019, pp. 384–389.

[21] 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.

[22] R. Schmucker, J. Wang, S. Hu, and T. M. Mitchell, "Assessing the performance of online students–new data, new approaches, improved accuracy," *J. Educ. Data Mining*, vol. 14, no. 1, pp. 1–45, 2022.

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

[24] L. Razzaq et al., "The ASSISTment builder: Supporting the life cycle of tutoring system content creation," *IEEE Trans. Learn. Technol.*, vol. 2, no. 2, pp. 157–166, Apr.–Jun. 2009.

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

[26] 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.

[27] 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. Anal. Knowl. Conf.*, 2021, pp. 490–496.

[28] S. Shen et al., "Monitoring student progress for learning process-consistent knowledge tracing," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8213–8227, Aug. 2023.

[29] J. Zhu, X. Ma, and C. Huang, "Stable knowledge tracing using causal inference," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 124–134, Apr. 2023. [Online]. Available: https://doi.org/10.1109/TLT.2023.3264772

[30] R. Pelánek, "Bayesian knowledge tracing, logistic models, and beyond: An overview of learner modeling techniques," *User Model. User-Adapted Interact.*, vol. 27, no. 3–5, pp. 313–350, 2017.

[31] 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.

[32] T. Liu, "Knowledge tracing: A bibliometric analysis," *Comput. Educ.: Artif. Intell.*, vol. 3, 2022, Art. no. 100090.

[33] A. Zanellati, D. D. Mitri, M. Gabbrielli, and O. Levrini, "Hybrid models for knowledge tracing: A systematic literature review," *IEEE Trans. Learn. Technol.*, vol. 17, pp. 1021–1036, Jan. 2024. [Online]. Available: https://doi.org/10.1109/TLT.2023.3348690

[34] P. I. Pavlik, L. G. Eglington, and L. M. Harrell-Williams, "Logistic knowledge tracing: A constrained framework for learner modeling," *IEEE Trans. Learn. Technol.*, vol. 14, no. 5, pp. 624–639, Oct. 2021.

[35] T. Käser, S. Klingler, A. G. Schwing, and M. Gross, "Dynamic Bayesian networks for student modeling," *IEEE Trans. Learn. Technol.*, vol. 10, no. 4, pp. 450–462, Oct.–Dec. 2017.

[36] J.-J. Vie and H. Kashima, "Knowledge tracing machines: Factorization machines for knowledge tracing," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 750–757.

[37] 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. Steering Committee*, 2017, pp. 765–774.

[38] X. Huang et al., "Constructing educational concept maps with multiple relationships from multi-source data," in *Proc. IEEE Int. Conf. Data Mining*, 2019, pp. 1108–1113.

[39] N. Thai-Nghe, L. Drumond, T. Horváth, A. Krohn-Grimberghe, A. Nanopoulos, and L. Schmidt-Thieme, "Factorization techniques for predicting student performance," in *Educational Recommender Systems and Technologies: Practices and Challenges*. IGI Global, 2012, pp. 129–153.

[40] N. Thai-Nghe, L. Drumond, T. Horváth, and L. Schmidt-Thieme, "Using factorization machines for student modeling," in *Proc. Int. Conf. User Model., Adaptat., Personalization*, 2012.

[41] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.

[42] M. Khajah, R. V. Lindsey, and M. C. Mozer, "How deep is knowledge tracing?," 2016, *arXiv:1604.02416*.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] 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.

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

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

[46] A. Graves et al., "Hybrid computing using a neural network with dynamic external memory," *Nature*, vol. 538, no. 7626, pp. 471–476, 2016.

[47] 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.

[48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North*, 2019, pp. 4171–4186.

[49] T. Brown et al., "Language models are few-shot learners," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.

[50] X. Wang, X. Mei, Q. Huang, Z. Han, and C. Huang, "Fine-grained learning performance prediction via adaptive sparse self-attention networks," *Inf. Sci.*, vol. 545, pp. 223–240, 2021.

[51] J. Zhu, W. Yu, Z. Zheng, C. Huang, and G. P. C. Fung, "Learning from interpretable analysis: Attention-based knowledge tracing," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2020, pp. 364–368.

[52] F. M. Lord, *Applications of Item Response Theory to Practical Testing Problems*. Hillsdale, NJ, USA: Lawrence Erlbaum, 1980.

[53] S. Pu and L. Becker, "Self-attention in knowledge tracing: Why it works," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2022, pp. 731–736.

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

[55] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2020.

[56] K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[57] Y. Lu, P. Chen, Y. Pian, and V. W. Zheng, "CMKT: Concept map driven knowledge tracing," *IEEE Trans. Learn. Technol.*, vol. 15, no. 4, pp. 467–480, Aug. 2022.

[58] S. Tong et al., "Structure-based knowledge tracing: An influence propagation view," in *Proc. IEEE 19th Int. Conf. Data Mining*, 2020, pp. 541–550.

[59] J. M. Royer, "Theories of the transfer of learning," *Educ. Psychol.*, vol. 14, no. 1, pp. 53–69, 1979.

[60] T. Long et al., "Automatical graph-based knowledge tracing," in *Proc. 15th Int. Conf. Educ. Data Mining*, 2022, pp. 710–714.

[61] 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.

[62] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, "Layer-wise relevance propagation: An overview," in *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Berlin, Germany: Springer, 2019, pp. 193–209.

[63] W. Saeed and C. Omlin, "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities," *Knowl.-Based Syst.*, vol. 263, 2023, Art. no. 110273.

[64] Y. Lu, D. Wang, P. Chen, Q. Meng, and S. Yu, "Interpreting deep learning models for knowledge tracing," *Int. J. Artif. Intell. Educ.*, vol. 33, no. 3, pp. 519–542, 2023.

[65] Q. Liu, Y. Ge, Z. Li, E. Chen, and H. Xiong, "Personalized travel package recommendation," in *Proc. IEEE 11th Int. Conf. Data Mining*, 2011, pp. 407–416.

[66] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized Bayesian knowledge tracing models," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer, 2013, pp. 171–180.

[67] Z. A. Pardos and N. T. Heffernan, "Modeling individualization in a Bayesian networks implementation of knowledge tracing," in *User Modeling, Adaptation, and Personalization*. Berlin, Germany: Springer, 2010, pp. 255–266.

[68] J. I. Lee and E. Brunskill, "The impact on individualizing student models on necessary practice opportunities," *Int. Educ. Data Mining Soc.*, 2012.

[69] Z. A. Pardos, S. Trivedi, N. T. Heffernan, and G. N. Sárközy, "Clustered knowledge tracing," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2012, pp. 405–410.

[70] S. Trivedi, Z. A. Pardos, and N. T. Heffernan, "Clustering students to generate an ensemble to improve standard test score predictions," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2011, pp. 377–384.

[71] Y. Yang, F. Shen, Z. Huang, H. T. Shen, and X. Li, "Discrete nonnegative spectral clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 9, pp. 1834–1845, Sep. 2017.

[72] B. Jiang, J. Pei, Y. Tao, and X. Lin, "Clustering uncertain data based on probability distribution similarity," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 4, pp. 751–763, Apr. 2013.

[73] 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. IEEE Int. Conf. Data Mining*, 2018, pp. 1182–1187.

[74] S. Mao, J. Zhan, Y. Wang, and Y. Jiang, "Improving knowledge tracing via considering two types of actual differences from exercises and prior knowledge," *IEEE Trans. Learn. Technol.*, vol. 16, no. 3, pp. 324–338, Jun. 2023.

[75] Y. Zhao, H. Ma, W. Wang, W. Gao, F. Yang, and X. He, "Exploiting multiple question factors for knowledge tracing," *Expert Syst. Appl.*, vol. 223, 2023, Art. no. 119786.

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

[77] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.

[78] V. Trowler, "Student engagement literature review," *Higher Educ. Acad.*, vol. 11, no. 1, pp. 1–15, 2010.

[79] R. M. Carini, G. D. Kuh, and S. P. Klein, "Student engagement and student learning: Testing the linkages," *Res. Higher Educ.*, vol. 47, no. 1, pp. 1–32, 2006.

[80] C. Berka et al., "EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks," *Aviation, Space, Environ. Med.*, vol. 78, no. 5, pp. B231–B244, 2007.

[81] Y. Xu, K.-M. Chang, Y. Yuan, and J. Mostow, "EEG helps knowledge tracing!," in *Proc. Workshop Utilizing EEG Input Intell. Tutoring Syst.*, 2014, Art. no. 43.

[82] S. Schultz and I. Arroyo, "Tracing knowledge and engagement in parallel in an intelligent tutoring system," in *Proc. Educ. Data Mining*, 2014, pp. 312–315.

[83] S. E. Schultz and I. Arroyo, "Modeling the interplay between knowledge and affective engagement in students," *Int. J. People-Oriented Program.*, vol. 3, no. 2, pp. 56–74, 2014.

[84] K. Mongkhonvanit, K. Kanopka, and D. Lang, "Deep knowledge tracing and engagement with MOOCs," in *Proc. 9th Int. Conf. Learn. Anal. Knowl.*, 2019, pp. 340–342.

[85] S. Markovitch and P. D. Scott, "The role of forgetting in learning," in *Proc. Int. Conf. Mach. Learn.*, 1988, pp. 459–465.

[86] Y. Chen et al., "Tracking knowledge proficiency of students with educational priors," in *Proc. 26th ACM Int. Conf. Inf. Knowl. Manage.*, 2017, pp. 989–998.

[87] 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.

[88] 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. Educ. Data Mining*, 2011, pp. 139–148.

[89] P. Nedungadi and M. Remya, "Incorporating forgetting in the personalized, clustered, Bayesian knowledge tracing (PC-BKT) model," in *Proc. Int. Conf. Cogn. Comput. Inf. Process.*, 2015, pp. 1–5.

[90] G. R. Loftus, "Evaluating forgetting curves," *J. Exp. Psychol.: Learn., Memory, Cogn.*, vol. 11, no. 2, 1985, Art. no. 397.

[91] R. Pelánek, J. Papoušek, and V. Stanislav, "Adaptive practice of facts in domains with varied prior knowledge," in *Proc. 7th Int. Conf. Educ. Data Mining*, 2014, pp. 6–13.

[92] R. Pelánek, "Application of time decay functions and ELO system in student modeling," in *Proc. Educ. Data Mining*, 2014, pp. 21–27.

[93] R. Pelánek, "Modeling students' memory for application in adaptive educational systems," in *Proc. 8th Int. Educ. Data Mining Soc.*, 2015, pp. 480–483.

[94] 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.

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

[96] Y. Wang and N. T. Heffernan, "Leveraging first response time into the knowledge tracing model," in *Proc. Int. Educ. Data Mining Soc.*, 2012, pp. 176–179.

[97] J. E. Beck, K.-M. Chang, J. Mostow, and A. Corbett, "Does help help? Introducing the Bayesian evaluation and assessment methodology," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2008, pp. 383–394.

[98] M. Sao Pedro, R. Baker, and J. Gobert, "Incorporating scaffolding and tutor context into Bayesian knowledge tracing to predict inquiry skill acquisition," *Educ. Data Mining*, 2013.

[99] C. Lin and M. Chi, "Intervention-BKT: Incorporating instructional interventions into Bayesian knowledge tracing," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2016, pp. 208–218.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

[100] J. González-Brenes, Y. Huang, and P. Brusilovsky, "General features in knowledge tracing to model multiple subskills, temporal item response theory, and expert knowledge," in *Proc. 7th Int. Conf. Educ. Data Mining*, 2014, pp. 84–91.

[101] T. Berg-Kirkpatrick, A. Bouchard-Côté, J. DeNero, and D. Klein, "Painless unsupervised learning with features," in *Proc. Annu. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, 2010, pp. 582–590.

[102] 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.

[103] 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.

[104] Z. Huang et al., "Question difficulty prediction for reading problems in standard tests," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1352–1359.

[105] Q. Liu et al., "Finding similar exercises in online education systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1821–1830.

[106] 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.

[107] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[108] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev.: Data Mining Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011.

[109] L. P. Cheung and H. Yang, "Heterogeneous features integration in deep knowledge tracing," in *Proc. Int. Conf. Neural Inf. Process.*, 2017, pp. 653–662.

[110] H. Jung, J. Yoo, Y. Yoon, and Y. Jang, "Language proficiency enhanced knowledge tracing," in *Proc. Int. Conf. Intell. Tutoring Syst.*, 2023, pp. 3–15.

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

[112] J. Kasurinen and U. Nikula, "Estimating programming knowledge with Bayesian knowledge tracing," *ACM SIGCSE Bull.*, vol. 41, no. 3, pp. 313–317, 2009.

[113] L. Wang, A. Sy, L. Liu, and C. Piech, "Deep knowledge tracing on programming exercises," in *Proc. 4th ACM Conf. Learn., Scale*, 2017, pp. 201–204.

[114] M. Zhu, S. Han, P. Yuan, and X. Lu, "Enhancing programming knowledge tracing by interacting programming skills and student code," in *Proc. 12th Int. Learn. Anal. Knowl. Conf.*, 2022, pp. 438–443.

[115] M. C. Desmarais and R. S. Baker, "A review of recent advances in learner and skill modeling in intelligent learning environments," *User Model. User-Adapted Interact.*, vol. 22, no. 1, pp. 9–38, 2012.

[116] H. Wan, Z. Zhong, L. Tang, and X. Gao, "Pedagogical interventions in SPOCs: Learning behavior dashboards and knowledge tracing support exercise recommendation," *IEEE Trans. Learn. Technol.*, vol. 16, no. 3, pp. 431–442, Jun. 2023.

[117] Z. Machardy, "Applications of Bayesian knowledge tracing to the curation of educational videos," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA USA, Tech. Rep. UCB/EECS-2015-98, May 2015.

[118] Z. Huang, Q. Liu, C. Zhai, Y. Yin, and G. Hu, "Exploring multi-objective exercise recommendations in online education systems," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 1261–1270.

[119] S. Oxman and W. Wong, "White paper: Adaptive learning systems," DV X Innovations DeVry Education Group, pp. 1–30, Feb. 2014.

[120] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, "Player modeling," *Dagstuhl Follow UPS*, 2013.

[121] S. M. Fisch, R. Lesh, E. Motoki, S. Crespo, and V. F. Melfi, "Children's mathematical reasoning in online games: Can data mining reveal strategic thinking?," *Child Develop. Perspectives*, vol. 5, no. 2, pp. 88–92, 2011.

[122] P. Kantharaju, K. Alderfer, J. Zhu, B. Char, B. Smith, and S. Ontaón, "Tracing player knowledge in a parallel programming educational game," in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2019, pp. 173–179.

[123] Y. Long and V. Aleven, "Educational game and intelligent tutoring system: A classroom study and comparative design analysis," *ACM Trans. Comput.- Hum. Interact.*, vol. 24, no. 3, pp. 1–27, 2017.

[124] Z. Wang, H. Sun, and T. Han, "Predicting crowdsourcing worker performance with knowledge tracing," in *Proc. 13th Int. Conf. Knowl. Sci., Eng. Manage.*, 2020, pp. 352–359.

[125] S. Abdi, H. Khosravi, and S. Sadiq, "Modelling learners in crowdsourcing educational systems," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, 2020, pp. 3–9.

[126] K. Crowston et al., "Knowledge tracing to model learning in online citizen science projects," *IEEE Trans. Learn. Technol.*, vol. 13, no. 1, pp. 123–134, Jan.–Mar. 2020.

[127] T.-Y. Zhao, M. Zeng, and J.-H. Feng, "An exercise collection auto-assembling framework with knowledge tracing and reinforcement learning," *J. Comput. Sci. Technol.*, vol. 37, no. 5, pp. 1105–1117, 2022.

[128] Y. Shang et al., "Reinforcement learning guided multi-objective exam paper generation," in *Proc. SIAM Int. Conf. Data Mining*, 2023, pp. 829–837.

[129] M. Feng, N. Heffernan, and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *USER-ADAP*, vol. 19, no. 3, pp. 243–266, 2009.

[130] Y. Wang, N. T. Heffernan, and C. Heffernan, "Towards better affect detectors: Effect of missing skills, class features and common wrong answers," in *Proc. 5th Int. Conf. Learn. Analytics Knowl.*, 2015, pp. 31–35.

[131] H. Chang, H. Hsu, and K. Chen, "Modeling exercise relationships in e-learning: A unified approach," in *Proc. Int. Educ. Data Mining Soc.*, 2015, pp. 532–535.

[132] Z. Wang et al., "Diagnostic questions: The NeurIPS 2020 education challenge," 2020, *arXiv:2007.12061*.

[133] K. R. Koedinger, R. S. Baker, K. Cunningham, A. Skogsholm, B. Leber, and J. Stamper, "A data repository for the EDM community: The PSLC DataShop," in *Handbook of Educational Data Mining*. Boca Raton, FL, USA: CRC Press, 2010, pp. 65–78.

[134] Y. Choi et al., "EDNet: A large-scale hierarchical dataset in education," in *Proc. 21st Int. Conf. Artif. Intell. Educ.*, 2019, pp. 69–73.

[135] S. H. Edwards and K. P. Murali, "CodeWorkout: Short programming exercises with built-in data collection," in *Proc. ACM Conf. Innov. Technol. Comput. Sci. Educ.*, 2017, pp. 188–193.

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

[137] Q. Li et al., "A genetic causal explainer for deep knowledge tracing," *IEEE Trans. Evol. Comput.*, early access, Jun 15, 2023, doi: 10.1109/TEVC.2023.3286666.

[138] S. Minn, J.-J. Vie, K. Takeuchi, H. Kashima, and F. Zhu, "Interpretable knowledge tracing: Simple and efficient student modeling with causal relations," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 12810–12818.

[139] Y. Yang et al., "GIKT: A graph-based interaction model for knowledge tracing," in *Proc. Mach. Learn. Knowl. Discov. Databases: Eur. Conf.*, 2020, 2021, pp. 299–315.

[140] 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.

[141] 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.

[142] H. Dai, Y. Yun, Y. Zhang, W. Zhang, and X. Shang, "Contrastive deep knowledge tracing," in *Proc. Int. Conf. Artif. Intell. Educ.*, 2022, pp. 289–292.

[143] 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.

[144] N. Liu, Z. Wang, R. Baraniuk, and A. Lan, "Open-ended knowledge tracing for computer science education," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2022, pp. 3849–3862.

[145] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," *Learn. Individual Differences*, vol. 103, 2023, Art. no. 102274.

[146] E. Vasilyeva, S. Puuronen, M. Pechenizkiy, and P. Rasanen, "Feedback adaptation in web-based learning systems," *Int. J. Continuing Eng. Educ. Life Long Learn.*, vol. 17, no. 4/5, 2007, Art. no. 337.

[147] M. Wang, Z. K. Yang, S. Y. Liu, H. N. H. Cheng, and Z. Liu, "Using feedback to improve learning: Differentiating between correct and erroneous examples," in *Proc. Int. Symp. Educ. Technol.*, 2015, pp. 99–103.

[148] G. Fischer, "User modeling in human–computer interaction," *User Model. User-Adapted Interact.*, vol. 11, no. 1/2, pp. 65–86, 2001.

[149] W. X. Zhao et al., "A survey of large language models," 2023, *arXiv:2303.18223*.

[150] X. Pan, M. Zhang, S. Ji, and M. Yang, "Privacy risks of general-purpose language models," in *Proc. IEEE Symp. Secur. Privacy*, 2020, pp. 1314–1331.

[151] M. Wotto, "The future high education distance learning in Canada, the United States, and France: Insights from before COVID-19 secondary data analysis," *J. Educ. Technol. Syst.*, vol. 49, no. 2, pp. 262–281, 2020.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.

Shuanghong Shen received the B.E. degree in electronic engineering from Wuhan University, Wuhan, China, in 2018. He is currently working toward the Ph.D. degree in computer science with the University of Science and Technology of China, Hefei, China.

He has authored or coauthored papers published in referred conference proceedings, such as KDD2023, AAAI2022, and SIGIR2022. His research interests include data mining, knowledge discovery, natural language processing, and intelligent tutoring systems.

Dr. Shen was the recipient of the first prize in task 2 of the NeurIPS 2020 Education Challenge.

Qi Liu (Member, IEEE) received the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2013.

He is currently a Professor with the School of Computer Science and Technology, USTC. He has authored or coauthored papers published prolifically in refereed journals and conference proceedings, including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ACM Transactions on Information Systems, and KDD conference. His research interests include data mining and intelligent education.

Dr. Liu is an Associate Editor for IEEE TRANSACTIONS ON BIG DATA and *Neurocomputing*. He was the recipient of KDD' 18 Best Student Paper Award, ICDM' 11 Best Research Paper Award, and China Outstanding Youth Science Foundation in 2019. He is a Member of the Alibaba DAMO Academy Young Fellow.

Zhenya Huang (Member, IEEE) received the B.E. degree in software engineering from Shandong University, Jinan, China, in 2014, and the Ph.D. degree in computer science from the University of Science and Technology of China (USTC), Hefei, China, in 2020.

He is currently an Associate Researcher with the School of Computer Science and Technology, USTC. He has authored or coauthored more than 30 papers in refereed journals and conference proceedings including IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ACM Transactions on Information Systems, KDD, AAAI. His research interests include data mining, knowledge discovery representation learning, and intelligent tutoring systems.

Dr. Huang has served regularly in the program committees of a number of conferences, and is Reviewer for the leading academic journals.

Yonghe Zheng received the B.S. degree in mathematics from Sichuan Normal University, Chengdu, China, and the M.S. degree in cybernetics from the Harbin University of Science and Technology, Harbin, China.

He is currently the Funding Dean of the Research Institute of Science Education, Beijing Normal University, Beijing, China, where he is also a Professor and Ph.D. Supervisor. In 2018, he joined the Educational Faculty of Normal University, with specific research interests on science education and educational technology. He has established the Research Institute of Science Education of Beijing Normal University in 2019. He is engaged in project management of basic research and S&T policy research for a long time. He has authored or coauthored than 60 articles.

Prof. Zheng is the Vice president of the 8th Council of China Association of Children's Science Instructors; the Member of Teaching Guidance Committee of the Basic Education, Comprehensive Group of the Compulsory Education Curriculum Revision Committee, and Science Curriculum Standard Revision Group; the General Secretary of the Awarding Scheme Council for outstanding computer teachers in universities; the Vice Director of the Academic Committee of the Chinese Society of Natural Science Museum; and former Director General of the Science Policy Bureau of National Natural Science Foundation of China. He is the Chief Editor of *Disciplinary and Interdisciplinary Science Education Research Journal*.

Minghao Yin (Member, IEEE) received the B.S. and M.S. degrees from Northeast Normal University, Changchun, China, in 2001 and 2004, respectively, and the Ph.D. degree from Jilin University, Changchun, China, in 2008, all in computer science.

He is currently a Professor, the Dean of the School of Information Science and Information Technology, Northeast Normal University, and the Director of the Key Laboratory of Intelligent Information Processing in Jilin province colleges and universities. He has authored or coauthored multiple papers published in prestigious journals and conferences, including Artificial Intelligence, IEEE transactions, ACM transactions, JAIR, AAAI, IJCAI, and WWW. His research interests include heuristic search, data mining, and combinatorial optimization.

Dr. Yin is an Associate Editor for IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES (TLT)..

Minjuan Wang (Member, IEEE) received the B.A. degree in Chinese literature from the Beijing/Peking University, Beijing, China, in 1995, the M.A. degree in comparative literature from Penn State University, State College, PA, USA, in 1997, and the Ph.D. degree in information science and learning technologies from the University of Missouri, Columbia, MO, USA in 2001.

She is a newly appointed Chair Professor of emerging technologies and future education with The Education University of Hong Kong, Hong Kong. Since 2000, she has also been a Professor and Program Chair of learning design and technology with San Diego State University, San Diego, CA, USA. Her research interests include STEM education, new and emerging technologies in various educational settings, metaverse and immersive learning, and the design and implementation of artificial intelligence including AIGC for education and training.

Dr. Wang is the Editor-in-Chief of IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES (TLT). She is an internationally recognized scholar and has keynoted about 45 international conferences. She is recognized internationally for her research, publishing and dedicated service to IEEE and other scholarly communities. He is a Member of IEEE Education Society. She also the Co-Chair of the Education Society's technical committee for immersive learning (TC-ILE).

Enhong Chen (Fellow, IEEE) received the B.S. degree from Anhui University, Hefei, China, in 1989, the M.S. degree from the Hefei University of Technology, Hefei, in 1992, and the Ph.D. degree from the University of Science and Technology of China (USTC), Hefei, in 1996, respectively, all in computer science.

He is currently a Professor and the Executive Dean of the School of Data Science and the Director of the Anhui Province Key Laboratory of Big Data Analysis and Application. He has authored or coauthored a number of papers on refereed journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ACM Transactions on Intelligent Systems and Technology, IEEE TRANSACTIONS ON MOBILE COMPUTING, KDD, ICDM, NIPS, and CIKM. His current research interests include data mining and machine learning, social network analysis, and recommender system.

Dr. Chen was the recipient of the National Science Fund for Distinguished Young Scholars of China, Best Application Paper Award on KDD-2008, Best Student Paper Award on KDD-2018 (Research), and Best Research Paper Award on ICDM-2011. He is an ACM Distinguished Scientist.

Authorized licensed use limited to: University of Science & Technology of China. Downloaded on October 08,2024 at 08:40:17 UTC from IEEE Xplore. Restrictions apply.