

EduStudio: towards a unified library for student cognitive modeling

Le WU¹, Xiangzhi CHEN (✉)¹, Fei LIU (✉)¹, Junsong XIE¹, Chenao XIA¹, Zhengtao TAN¹,
Mi TIAN¹, Jinglong LI¹, Kun ZHANG¹, Defu LIAN², Richang HONG¹, Meng WANG¹

¹ Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei 230601, China

² School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China

© The Author(s) 2024. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract Student cognitive modeling is a fundamental task in the intelligence education field. It serves as the basis for various downstream applications, such as student profiling, personalized educational content recommendation, and adaptive testing. Cognitive Diagnosis (CD) and Knowledge Tracing (KT) are two mainstream categories for student cognitive modeling, which measure the cognitive ability from a limited time (e.g., an exam) and the learning ability dynamics over a long period (e.g., learning records from a year), respectively. Recent efforts have been dedicated to the development of open-source code libraries for student cognitive modeling. However, existing libraries often focus on a particular category and overlook the relationships between them. Additionally, these libraries lack sufficient modularization, which hinders reusability. To address these limitations, we have developed a unified PyTorch-based library EduStudio, which unifies CD and KT for student cognitive modeling. The design philosophy of EduStudio is from two folds. From a horizontal perspective, EduStudio employs the modularization that separates the main step pipeline of each algorithm. From a vertical perspective, we use templates with the inheritance style to implement each module. We also provide eco-services of EduStudio, such as the repository that collects resources about student cognitive modeling and the leaderboard that demonstrates comparison among models. Our open-source project is available at the website of edustudio.ai.

Keywords open-source library, student cognitive modeling, intelligence education

1 Introduction

Artificial Intelligence for Education (AI4ED) [1–3] has revolutionized traditional education and emerged as a trending topic. These modern educational platforms are dedicated to leveraging artificial intelligence technology to provide personalized and high-quality educational experiences for students. Student cognitive modeling [4], as a fundamental

task in the intelligent tutoring system, aims to capture students' cognitive ability on diverse aspects (typically various knowledge components (KC)) through their historical learning behaviors (especially exercise answering records). Accurate student cognitive modeling can facilitate a wide range of downstream tasks, such as student profiling [5,6], education resource recommendation [7–14], adaptive testing [15], and so on.

With the rapid progress of student cognitive modeling, it is imperative for researchers to develop a project for easily reproducing these published algorithms and designing new algorithms with minimum effort. However, this is not trivial as current student cognitive modeling works are rather fragmented. Researchers put repeated efforts into finding related datasets and reproducing algorithms. Therefore, there is a need to reconsider the implementation of student cognitive modeling techniques. In this paper, we develop a PyTorch-based library called EduStudio for student cognitive modeling and provide a range of user-friendly eco-services to enhance EduStudio. We are committed to promoting research and development for the AI4ED community.

EduStudio integrates models in both cognitive diagnosis (CD) [16,17] and knowledge tracing (KT) [18–20], which are two mainstream categories in student cognitive modeling field. Figure 1 shows the widely used application scenarios of these two categories. Specifically, CD is often used to quantify a student's cognitive ability (e.g., the mastery degree of a specific KC) with well-designed questions from an assessment or test. For instance, a well-known scenario of CD is the Programme for International Student Assessment (PISA) [21,22], with around 690,000 students took the PISA assessment in 2022, representing about 29 million 15-year-olds from schools in 81 participating countries and economies [23].

CD is based on the static cognitive assumption and deals with the challenge of how to better quantify student ability with sparse student records. In contrast, KT focuses on tracking students' knowledge states over a long period with the dynamic cognitive assumption. Many online tutoring APPs are equipped with KT technologies. As such, these APPs can

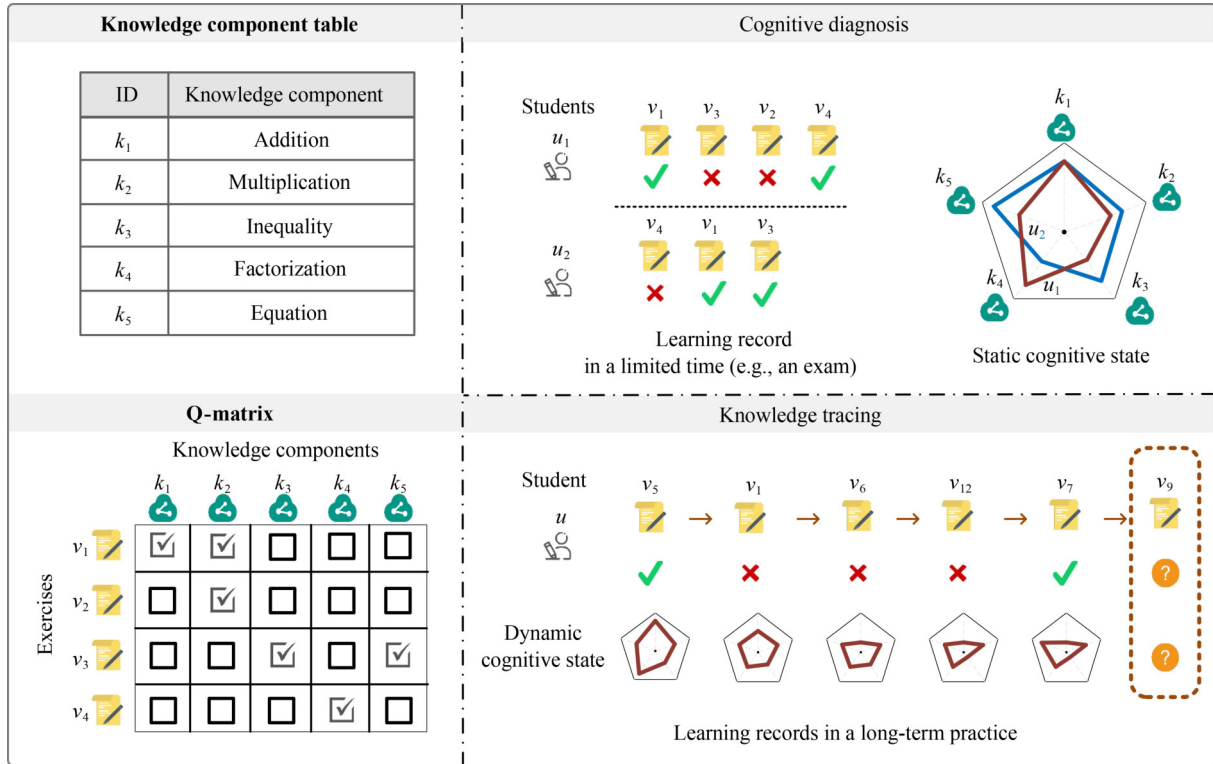


Fig. 1 Introduction to student cognitive modeling, including CD and KT

provide personalized exercise recommendation to improve student's abilities as they can see feedbacks after answering one exercise) and predict their future performances by mining their historical learning behaviors [7,24,25]. In summary, these two categories utilize student answering records to mine students' cognitive ability. However, due to the differences in cognitive modeling approaches between CD and KT, existing libraries often focus on a particular category and overlook the relationships between them.

Recent efforts have been devoted to developing libraries for student cognitive modeling [26–28]. They consider static or dynamic modeling separately and implement some cognitive modeling models. Nevertheless, we have identified some shortcomings and limitations in their endeavors to advance the community. Existing libraries: 1) Focus on a single category, which ignore the relation between the two categories. 2) Lack sufficient abstraction, which leads to poor flexibility and reusability. 3) Lack adequate eco-services, which limits the development of the community. Therefore, we desire to develop a high reusable and flexible library including CD and KT, with the comprehensive eco-services. The comparison of our EduStudio with other libraries is detailed in Section 7. The primary advanced features of our EduStudio are summarized as follows:

- We develop a unified library that combines the CD and KT under the student cognitive modeling view. Unlike existing open-source libraries that primarily focus on a single category, we not only enable the reusability within individual category but also facilitate sufficient reusability between two categories. We aim to facilitate communication between two research groups for better

student cognitive modeling.

- We provide the modularized and templated design when implementing models for better flexibility and reusability. Existing libraries often lack clear boundaries between the individual procedures in the algorithmic pipeline, leading to poor flexibility. We decompose each algorithm pipeline into six modules, and propose horizontal modularization flow of each algorithm. Besides, we extract the commonality of each module with reusable templates, and implement vertical templation design of each module for high-level management.
- We offer a range of eco-services surrounding EduStudio, which can further enable more researchers to understand and quickly participate in the field of student cognitive modeling. We provide a Github repository that collects valuable resources for student cognitive modeling. In addition, we develop a Leaderboard website to provide a comprehensive comparison of various models.

2 Background

In this section, we introduce the category and data description of student cognitive modeling. Subsequently, we provide a review of existing works including CD and KT.

2.1 Task and data description

Task description. Student cognitive modeling aims to model students' cognitive states based on learning data, such as their interactive records of answering exercises. Classified from the perspective of variation in cognitive states, CD and KT are two mainstream categories for modeling students' cognitive

states. CD is typically used to assess students' static cognitive states on knowledge components. It helps to understand students' knowledge mastery in specific domains and identify their weaknesses and areas for improvement. KT focuses more on monitoring students' dynamic cognitive changes and learning progress. It tracks the development of students' cognitive ability at different time steps and identifies their learning trajectories and trends. Therefore, CD and KT are two types of tasks proposed from different perspectives of cognitive state variation.

Data description. Here we discuss the various types of data involved in student cognitive modeling. As shown in Fig. 2, the dataset includes the interactive records of students answering exercises, as well as the relationship information between the exercises and the KCs. Additionally, the features of students and exercises, as well as the relations among KCs, also contain rich information that can enhance the accuracy of modeling. Various models selectively utilize different features and data formats based on their requirements.

- Student-side features typically include information about students' family background, school background, and other relevant factors. These pieces of information are valuable for modeling students' abilities as prior knowledge.
- Student-exercise interactions are the fundamental input for student cognitive modeling. It encompasses common features such as correctness labels, answering textual content, and interaction timestamp. In addition, some studies [29–31] also design diverse forgetting features via interaction timestamp to capture students' forgetting characteristics.
- Exercise-side features refer to the content information of exercises. This includes various modalities such as textual descriptions, images, and other multimedia elements associated with the exercises. They are valuable for modeling the difficulty of exercises and identifying the KCs they cover.
- Exercise-KC relationships are referred to as Q-matrix [32] in student cognitive modeling. Q-matrix reveals the KCs encompassed within each exercise. It serves as a bridge for establishing student cognition of KCs through student-exercise interactions.
- KC-side features mainly lies on KC relationships, which typically fall into two categories: inclusion relationships

and prerequisite relationships. Inclusion relationships refer to the coarse-grained KCs that encompass multiple finer-grained KCs. Prerequisite relationships indicate that one KC usually be learned before another.

2.2 Existing works

We introduce the development of student cognitive modeling, including CD and KT.

Cognitive diagnosis. Originating from psychometrics, CD emerges as a pivotal branch of test theory. Test theory methods are predominantly formulated on the foundations of educational and psychometric theories and assumptions [33]. The most exemplary of these is Item Response Theory (IRT) [34], which integrates factors such as student ability, exercise difficulty, exercise discrimination, and exercise guess probability into a logistic function to forecast the probability of a correct response. Like its predecessor, Classical Test Theory (CTT) [35], the student ability measured by IRT is on a macro level. Consequently, subsequent researchers propose the incorporation of micro-level knowledge structures (e.g., Q-matrix) into cognitive modeling [32,36,37], improving the interpretability of the model.

With the emergence of deep learning, the NCDM [38] model pioneers the use of neural networks to replace simple logical functions in modeling the complex interactions of students when answering exercises. Subsequently, more and more neural CD models [39–42] further refine the model architecture to enhance the prediction performance of CD. Beyond architectural enhancements, researchers are progressively integrating diverse data sources including exercise, student, and KC-side data. For exercise-side data, CNCD-F [39] and CNCD-Q [39] respectively extract the reading comprehension difficulty factor and KCs from the textual content of exercises. For student-side data, ECD [43] incorporates information such as the student's family background into the prediction process of student performance, while FairCD [44] and FairLISA [45] use student sensitive attributes for fairness research. Models like MGCD [46] utilize features such as the class identifier to consider group-level CD. Regarding KC-side data, RCD [47] and HierCDF [48] introduce the prerequisite relationships of KCs into CD to further enhance performance, while DCD [49] uses the inclusion relationships of KCs for CD in the scenario where there is substantial absence of KC annotations in exercises.

Knowledge tracing. In the field of KT, its early iterations primarily encompassed probabilistic models and logical models. Probabilistic models assume that the learning process follows a Markov process, where students' latent knowledge states can be estimated by their observed performance [19,50]. Within this paradigm, models such as BKT [51] and DBKT [52] stand out as exemplary. Logistic models constitute a significant category of models grounded in logistic functions, which encapsulate the probability of correctly answering exercises within a mathematical framework that accounts for both student and KC parameters. Notable models within this class include LFA [53], PFA [54], and KTM [55].

In the era of deep learning, the evolution of KT is

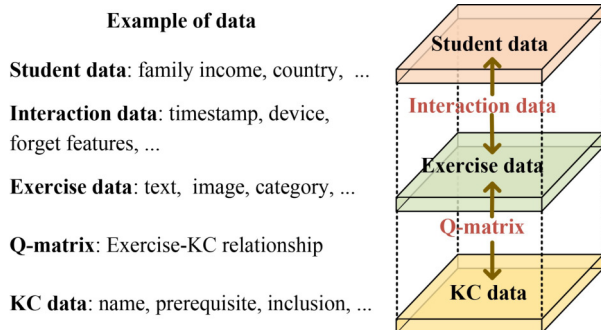


Fig. 2 Data description

manifested through the sophisticated network architectures to enhance performance. The primary characteristic lies in the incorporation of diverse network structures to model the dynamic cognition of students. DKT [24] pioneers the introduction of RNN and LSTM to model the evolving cognitive states of students. Subsequently, an array of models based on LSTM or RNN architectures have been proposed [29,56–58]. Inspired by memory-augmented neural networks, subsequent models begin to enhance the representation of students' memory processes [59–61]. With the rise of the transformer, there has been a surge in utilizing attention-based architectures [62–64]. Since the interactions between students and exercises, the relationships between exercises and KCs, and the interconnections among KCs can all be represented as graph structures, some researches explore graph-based KT [65,66].

Due to the similarities between CD and KT, some works that integrate CD and KT have also been proposed [67]. A typical category of such work is to use CD models to enhance the interpretability of traditional KT models [33,68,69]. For instance, Deep-IRT [68] is a synthesis of the IRT [34] model and DKVMN [59] to make deep learning-based KT interpretable. DynamicCD [33] incorporate educational priors from CD models into KT for better interpretability.

3 Overview of EduStudio

In this section, we first summarize the challenges faced in developing EduStudio when unifying CD and KT. To address these challenges, we present the design philosophy in Fig. 3. Grounded in the design philosophy, the overall architecture is depicted in Fig. 4.

3.1 Challenges of developing EduStudio

After introducing the background, we can observe that the data usage of student cognitive modeling is diverse and there are commonalities and differences in between CD and KT. Here we mainly analyzes the challenges of the process of developing a unified library for CD and KT. The solutions to these challenges are detailed in Section 4.7.

- **Unified management of multifaceted data.** Data utilized by CD and KT, relating to students, exercises,

and KCs, varies in format among different dataset publishers. Standardizing data file formats and maintaining commonality for effective data management is a pressing issue.

- **Ensuring reusability and flexibility in the context of unifying CD and KT.** Since both CD and KT are methods for student cognitive modeling, there are commonalities and differences in their approaches. Therefore, ensuring reusability for commonalities and ensuring flexibility for differences is a major challenge.
- **Compatibility for future task scenarios.** In both CD and KT, there are various task scenarios, such as fairness, cold start, and so on. When designing EduStudio, it is necessary to consider compatibility with both existing task scenarios and unknown future task scenarios.

3.2 Design philosophy

3.2.1 Horizontal modularization

From the horizontal modularization viewpoint, we decompose the general algorithmic pipeline into six modules: Configuration reading, data preparation, model implementation, training control, model evaluation, and log storage.

- Configuration reading (Step 1) aims to collect, categorize, and deliver configurations from different configuration portals.
- Data preparation (Step 2) aims to read raw data files from the disk and then convert them into model-friendly data objects.
- Model implementation (Step 3) refers to the process of implementing the structure of each model and facilitating the reuse of model components.
- Training control (Step 4) focuses on the training process of various models.
- Model evaluation (Step 5) focuses on the implementation of various evaluation metrics.
- Log storage (Step 6) aims to implement storage specifications when storing generated data.

Horizontal modularization establishes clear boundaries for

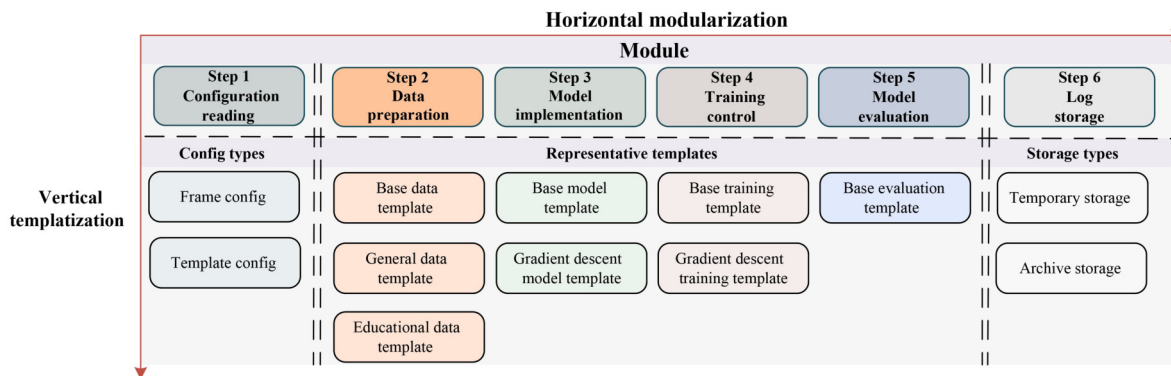


Fig. 3 EduStudio's design philosophy incorporates horizontal modularization and vertical templatzation to enhance flexibility and reusability. Horizontal modularization: We decompose the general algorithmic pipeline into six modules to enhance flexibility. Vertical templatzation: We implement reusable templates within the modules for Steps 2–5 to achieve high-level management of complex elements. Since all models share the same configuration reading method and log storage path management, there is no need for the template-based design for them

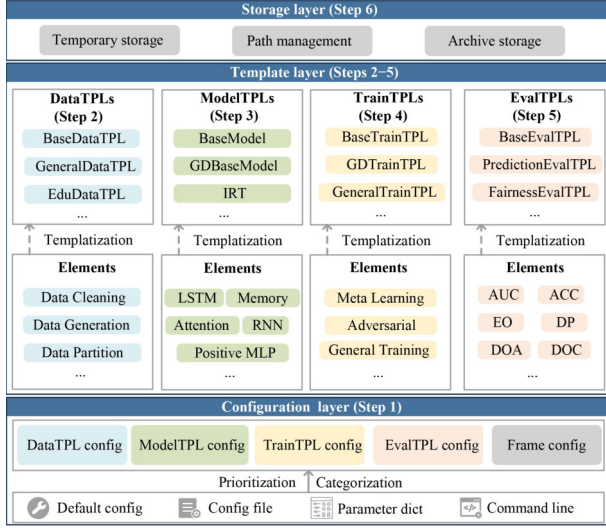


Fig. 4 The overall architecture of EduStudio

each step throughout the algorithm pipeline, facilitating the incorporation of new things to individual modules.

3.2.2 Vertical templatization

When it comes to a specific module, we observe that there are numerous elements within the module that require implementation and management. Without proper high-level management of these elements, subsequent development and reusability can become overly complex. Thus, we implement vertical templatization design within the modules for Steps 2–5 in Fig. 3. We manage these complex elements within the modules using templates, which ensures a well-organized structure. Furthermore, we have developed numerous base templates and created new templates by inheriting from these base templates. These templates are reusable by the models, enhancing their reusability. It should be noted that since all models share the same configuration reading method and log storage path management, these two modules are called in a common, model-independent area. In this case, there is no need for the template-based design for them.

3.3 Overall architecture

Based on the above design philosophy, the overall architecture of EduStudio is illustrated in Fig. 4. Steps 2–5 are four templatized modules, while Steps 1 and 6 are common modules that are shared by all the models.

For the four templatized modules (i.e., data preparation, model implementation, training control, and model evaluation), we abstract the intricate elements within each module into various reusable templates. Within each templatized module, we implement multiple templates with inheritance relationships. Each template inherits from a basic template prefixed with BaseBase. These basic templates only provide basic functionalities to maintain the fundamental operation of the library. With this templatized design, we can easily extend a new template within any module, enabling reusability when implementing new models.

In addition to the aforementioned four modules, there are two additional modules (i.e., configuration reading and log storage) that are shared by all models. For configuration

reading, we prioritize and categorize the configurations from four flexible configuration portals. This allows us to identify five categories of configurations, where four categories correspond to the four templatized modules, and the last category involves framework-specific configurations. For Log Storage, we store logs from failed or ongoing runs in temporary storage, while successful run logs are stored archivally. This allows users to conveniently discard failed experiments.

4 Design of EduStudio

We organize this section into multiple subsections based on horizontal modularization. Within each subsection, we delve into our vertical templatization design. Ultimately, we provide an in-depth explanation of addressing challenges that are described in Section 3.1.

4.1 Configuration reading

Configuration Reading aims to collect, categorize, and deliver configurations from different configuration portals. We first collect configurations from four flexible configuration portals (e.g., configuration file and command line). Then we retain the highest-priority configurations and categorize them into five groups: data template configuration, model template configuration, training template configuration, evaluation template configuration, and frame configuration (library-specific configurations). Categorized configuration objects make it easier for users to find and utilize them. Finally, we deliver categorized configuration objects to their corresponding modules.

4.2 Data preparation

Data Preparation aims to convert raw data from the hard disk into model-friendly data objects. Standardizing the data preparation pipeline is challenging in the library design because various student cognitive models utilize data with diverse content and formats. For example, CD handles interaction data ignoring timestamp, while KT handles sequential interaction data. Additionally, models may selectively utilize features such as relations, contexts, and other relevant data features.

To address the aforementioned challenges, let us first clarify the workflow of data preparation, as shown in Fig. 5. The first step is to load the raw data from the hard disk. Then, a series of processing steps are performed to obtain model-friendly data objects. Finally, these data objects are passed on to other modules. We simplify the data preparation into three stages:

- Data loading: Loading necessary data from the hard disk.
- Data processing: Convert the raw data into model-friendly data objects by a range of data processing operations.
- Data delivery: Deliver model-friendly data objects to other modules.

Among these three stages, data processing is the most complex and feature-rich stage in data preparation. Therefore, we have established a set of standardized protocols and

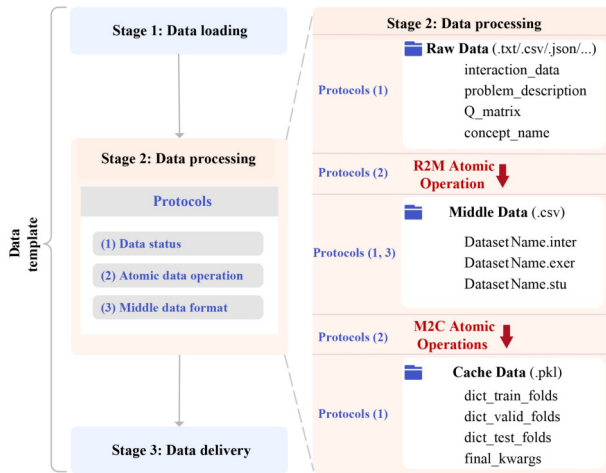


Fig. 5 Data Preparation includes the loading, processing, and delivery stages. We have established a set of standardized protocols and developed a series of atomic data operations for data processing (Section 4.2.1). We utilize data templates (Section 4.2.2) to manage and control the three stages, enabling reusable data preparation

developed a series of atomic data operations for data processing (detailed in Section 4.2.1). These protocols and operations help streamline and enhance the data processing stage, making it more efficient and effective. Finally, we utilize the data template (detailed in Section 4.2.2) to manage and control these three stages, enabling a complete and reusable data preparation process. The data template ensures consistency and standardization throughout the stages, facilitating efficient data preparation for the following steps.

4.2.1 Protocols for data processing

In order to standardize the complete workflow of data preparation, we propose three protocols for the data processing stage: data status, middle data format, and atomic data operation protocols.

- **Data status protocol.** We categorize data into three statuses: 1) *inconsistent rawdata*: the original data format provided by the dataset publisher. This data format is diverse and lacks unification; 2) *standardized middata*: the standardized middle data format defined by EduStudio. This unified format is friendly for researchers to read; 3) *model-friendly cachedata*: the data format that is convenient for model usage. In EduStudio, we implement data cache functionality, which allows users to bypass the data processing procedure in subsequent experiments after saving cached data from the previous experiment.
- **Middle data format protocol.** As mentioned in the Data Status Protocol, the middle data is the standardized data format. We define some standardized data files for student-exercise interaction data, student-side features, exercise-side features and so on, which is detailed in EduStudio official website.
- **Atomic data operation protocol.** To achieve reusability and flexibility in data preparation, we propose the concept of atomic data operation to convert the whole data processing into some reusable atomic data

operations. From rawdata to middata, we require users to specify one atomic data operation (i.e., a Python class prefixed with R2M) to convert raw data into standardized middata. From middata to cachedata, we allow users to specify multiple atomic data operations (i.e., multiple python classes prefixed with M2C) sequentially.

Founded on above protocols, we offer a comprehensive range of **atomic data operations** to facilitate the transformation of rawdata into middata, and subsequently into cachedata. These operations include *R2M* (Rawdata to Middata) and *M2C* (Middata to Cachedata) atomic operations. The flexibility to combine and substitute atomic operations enables flexibility.

- **Atomic data operations for transformation of rawdata to middata:** Due to the diverse nature of rawdata in different datasets, we provide a total of 18 *R2M* operations for all inherited datasets within the library. These operations are designed to transform the raw data into an intermediate data format, facilitating subsequent processing and analysis.
- **Atomic data operations for transformation of middata to cachedata:** To ensure the compatibility of data objects with models, particularly cachedata, we meticulously devise a range of *M2C* operations. These operations can be broadly classified into four main categories based on the type of data processing: data cleaning, data conversion, data partition, and data generation. As indicated in Table 1, data cleaning focuses on refining the data by applying filters to students or exercises and addressing missing values. Data conversion aims to modify the data format. We specifically design operations to accommodate the triple form in CD and the sequence form in KT. Data partition involves dividing the entire dataset into training, validation, and test sets for CD and KT. Data generation aims to produce additional features that can enhance prediction capabilities, such as KC inclusion relationships and KC prerequisite relationships.

4.2.2 Data templates

Data templates ensure consistency and standardization throughout the three stages of data preparation, facilitating efficient data preparation for the following steps. Table 2 demonstrates three highly reusable data templates: the base template, general template, and educational template. The base data template is not specific to educational data and provides basic functionalities to maintain the fundamental operation of the library. The general template inherits from the base template and focuses on scenarios involving simple educational data with only student-exercise interaction data. It implements three protocols in data preparation. The educational template inherits from the general template and includes additional student-side and exercise-side features. When implementing a new data template, the focus lies in loading data and composing various atomic data operations.

Table 1 Representative M2C atomic data operations that transform data from middata to cachedata

M2C operation type	M2C operation name	Description
Data cleaning	M2C_FilterRecords4CD	Filter some students or exercises according specific conditions
	M2C_Label2Int	Binarization for answering response
Data conversion	M2C_ReMapId	Identifier remapping of discrete features
	M2C_BuildSeqInterFeats	Build sample format for KT
Data partition	M2C_DataSplit4CD	Data partition for CD
	M2C_DataSplit4KT	Data partition for KT
Data generation	M2C_GenQMat	Generate Q-matrix
	M2C_BuildKCRelation	Build knowledge component relation graph

Table 2 Description of representative templates for four templatized modules in EduStudio

Template type	Template name	Parent template	Description
Data templates	BaseDataTPL	/	The basic class of data templates
	GeneralDataTPL	BaseDataTPL	Implement all protocols for data processing
	EduDataTPL	GeneralDataTPL	Load extra student-side and excise-side features based on GeneralDataTPL
Model templates	BaseModel	/	The basic class of model templates
	GDBaseModel	BaseModel	Provide utilities for gradient descent models based on BaseModel
Training templates	BaseTrainTPL	/	The basic class of training templates
	GDBaseTrainTPL	BaseTrainTPL	Provide utilities for gradient descent models based on BaseTrainTPL
	GeneralTrainTPL	GDBaseTrainTPL	The TrainTPL for general training
	AdversarialTrainTPL	GeneralTrainTPL	The TrainTPL for adversarial training
Evaluation templates	BaseEvalTPL	/	The basic class of evaluation templates
	PredictionEvalTPL	BaseEvalTPL	Student performance prediction evaluation
	InterpretabilityEvalTPL	BaseEvalTPL	Student cognitive representation interpretability evaluation
	IdentifiabilityEvalTPL	BaseEvalTPL	Student cognitive representation identifiability evaluation
	FairnessEvalTPL	BaseEvalTPL	Student cognitive fairness evaluation

4.3 Model implementation

Model Implementation refers to the process of implementing the structure of each model and facilitating the reuse of model components. We designed two basic model templates: Base (BaseModelTPL) and Gradient Descent Base (GDBaseModelTPL). By inheriting the basic model templates, we collectively implemented 45 student cognitive models.

As listed in Table 2, there are two basic model templates, namely BaseModelTPL and GDBaseModelTPL, which define the specifications for model implementation in EduStudio. The difference between BaseModelTPL and GDBaseModelTPL lies in the fact that the latter builds upon the former by considering models that can be optimized using gradient descent methods. GDBaseModelTPL provides additional tools and functionalities specifically designed for gradient descent-based optimization models. All models are required to inherit from one of these basic model templates and adhere to the corresponding interface functions. For example, we specify the interface function of *add_extra_data(.)* for loading extra required data (such as Q-matrix, KC relationships) except student-exercise interactions. Additionally, we define *get_loss_dict(.)* for returning a loss dictionary that contains multiple losses.

During the implementation process of the model, we develop reusable components for portability. For instance, we implement a Positive MultiLayer Perceptron (PosMLP) to support the monotonicity assumption [38] that is widely used in CD models for interpretability. The monotonicity assumption states that the probability of a correct response to an exercise increases monotonically with any dimension of the student’s cognitive proficiency.

We currently implement 16 models for CD and 29 models for KT in EduStudio. We arrange implemented models in terms of data usage and technique usage in Table 3.

4.4 Training control

Training control focuses on the training methods of different models. It is worth noting that in the training control procedure, some implemented training templates are shared between the CD and KT. This highlights the ability of EduStudio to promote significant reusability between them.

For the models that have been implemented so far, we summarize three mainstream training paradigms for student cognitive modeling and provide corresponding training template for each training paradigm: general training (GeneralTrainTPL) and adversarial training (AdversarialTrainTPL), as listed in Table 2. Their ancestral training template (i.e., BaseTrainTPL) provides the necessary functionality to maintain the basic operation of the library. GDBaseTrainTPL based on BaseTrainTPL provides some utilities for gradient descent based models. When a new training paradigm comes, we can inherit these base training templates to implement a new training template.

4.5 Model evaluation

Model evaluation primarily focuses on the implementation of various evaluation metrics. They can be shared by all CD and KT models according to their respective needs. As illustrated in Table 2, we currently implement four kinds of important metrics for student cognitive models.

- **Student performance prediction evaluation** aims to evaluate the prediction performance that students’

Table 3 Implemented 45 student cognitive models in EduStudio, including 16 CD models and 29 KT models

Category	Model	Publish	Year	Data	Technique
Static cognitive modeling (CD)	IRT [34]	–	1960	Interaction	IRT
	MIRT [70]	–	1982	Interaction	IRT
	DINA [32]	JEBS	2009	Interaction, Q-matrix	–
	NCDM [38]	AAAI	2020	Interaction, Q-matrix	MLP, IRT
	CDGK [40]	CIKM	2021	Interaction, Q-matrix	MLP, IRT
	MGCD [46]	ICDM	2021	Interaction, Q-matrix, Student Features	Attention
	RCD [47]	SIGIR	2021	Interaction, Q-matrix, KC Prerequisite Relationships	Graph Neural Network
	ECD [43]	SIGKDD	2021	Interaction, Q-matrix, Student Features	Hierarchical Attention
	CNCD-Q [39]	TKDE	2022	Interaction, Q-matrix	NCDM
	CNCD-F [39]	TKDE	2022	Interaction, Q-matrix, Exercise Texts	TextCNN, NCDM
	KaNCD [39]	TKDE	2022	Interaction, Q-matrix	NCDM
	KSCD [41]	CIKM	2022	Interaction, Q-matrix	NCDM
	CDMFKC [42]	CIKM	2022	Interaction, Q-matrix	NCDM
	HierCDF [48]	SIGKDD	2022	Interaction, Q-matrix, KC Prerequisite Relationships	Bayesian Network
	FairCD [44]	SCIS	2023	Interaction, Q-matrix, Student Features	Disentanglement, Adversarial
	DCD [49]	NeurIPS	2023	Interaction, Q-matrix, KC Inclusion Relationships	Disentanglement, VAE
Dynamic cognitive modeling (KT)	DKT [24]	NeurIPS	2015	Interaction	RNN/LSTM
	DKVMN [59]	WWW	2017	Interaction	Memory
	DKT_DSC [57]	ICDM	2018	Interaction	RNN/LSTM
	EERNN [71]	AAAI	2018	Interaction	LSTM, Attention
	DKT+ [56]	L@S	2018	Interaction	RNN/LSTM
	SAKT [62]	EDM	2019	Interaction	Attention
	SKVMN [60]	SIGIR	2019	Interaction	Memory
	Deep-IRT [68]	EDM	2019	Interaction	Memory, IRT
	KQN [72]	LAK	2019	Interaction	GRU/LSTM
	DKTForget [29]	WWW	2019	Interaction, Q-matrix	RNN/LSTM
	GKT [65]	WI	2019	Interaction	Graph Neural Network
	EKT [61]	TKDE	2019	Interaction, Q-matrix, Exercise Texts	LSTM, Attention, Memory
	qDKT [58]	EDM	2020	Interaction	RNN/LSTM
	AKT [73]	SIGKDD	2020	Interaction, Q-matrix	Attention
	CKT [74]	SIGIR	2020	Interaction	CNN
	RKT [75]	CIKM	2020	Interaction, Exercise Relation Graph	Attention
	SAINT [63]	L@S	2020	Interaction, Exercise Features	Attention, Transformer
	SAINT+ [64]	LAK	2021	Interaction, Exercise Features	Attention, Transformer
	ATKT [76]	ACM MM	2021	Interaction, Q-matrix	Attention, LSTM
	IEKT [77]	SIGIR	2021	Interaction, Q-matrix	GRU
	LPKT [30]	SIGKDD	2021	Interaction, Q-matrix	GRU, MLP
	HawkesKT [78]	WSDM	2021	Interaction, Q-matrix	Hawkes Process
	CT-NCM [79]	IJCAI	2022	Interaction, Q-matrix	Hawkes Process, LSTM
	LPKT-S [31]	TKDE	2022	Interaction, Q-matrix	GRU, MLP
	CL4KT [80]	WWW	2022	Interaction, Q-matrix	Transformer, Contrastive Learning
	DIMKT [81]	SIGIR	2022	Interaction, Q-matrix	Sequential Neural Network
	QIKT [82]	AAAI	2023	Interaction, Q-matrix	LSTM, IRT
	SimpleKT [83]	ICLR	2023	Interaction, Q-matrix	Attention
	DTransformer [84]	WWW	2023	Interaction, Q-matrix	Transformer, Contrastive Learning

response to exercises, which usually can be formulated as a binary classification task. Common metrics include classification metrics such as Area Under the Curve (AUC) and ACCuracy (ACC), as well as regression metrics like Root Mean Square Error (RMSE).

- **Cognitive representation interpretability evaluation** aims to evaluate the students' cognitive results. NCDM [38] proposes the Degree of Agreement (DOA) metric whose intuition is that if student a has a better mastery on KC k than student b , then a is more likely to answer exercises related to k correctly than b . The authors of IC-IDM [85] consider that the order of interpretable students' knowledge proficiencies should be consistent with the order of response scores on relevant exercises.

They propose the Degree of Consistency (DOC) metric.

- **Cognitive representation identifiability evaluation** aims to measure the discrepancy between cognitive ability of students with the same response distribution. In general, students exhibiting the same response distribution should demonstrate similar cognitive outcomes. IC-IDM [85] proposes the identifiability concept of various CD models and a quantitative Identifiability Score (IDS) to measure the identifiability.
- **Cognitive fairness evaluation** aims to measure the fairness. FairCD [44] explores the fairness in CD and proposes the F_{CD} metric whose intuition is that a model is considered to be fair if the gap between true proficiency and predicted proficiency is identical across

different groups. FairLISA [45] utilizes the classical fairness metrics: Demographic Parity (DP) [86] and Equal Opportunity (EO) [87] to measure the fairness.

4.6 Log storage

Log Storage aims to implement storage specification when storing generated data primarily depends on path management. Table 4 displays the path management¹⁾. For path management of log storage, we specify `<project>/temp/` directory to store logs of ongoing or failed experiments as temporary storage and `<project>/archive/` directory to store logs of completed experiments as archive storage, which is convenient for users to abandon failed experiments. When it comes to a detailed experiment log, we stipulate: 1) `config.json`: store all configuration information; 2) `<ID>.log`: store training log; 3) `result.json`: store model evaluation result; 4) `/pth/`: store model parameters at each epoch or the best epoch.

4.7 Summary

After introducing the detailed design of EduStudio, in this section, we elaborate our solutions to the challenges discussed in Section 3.1.

The primary challenge of the EduStudio is to efficiently reuse the commonalities (reusability) of CD and KT while preserving their differences (flexibility). We adopt a modularized and templated design (detailed in Section 3.2) to address this challenge. This design philosophy is reflected in all six delineated modules, which is the content that this subsection will elaborate on. 1) For the modules of Configuration Reading and Log Storage, we reuse the same configuration and storage methodologies across both the CD and KT, as these two modules are task-agnostic. 2) In the Data Preparation module, we segment the entire data processing process into a series of atomic data operations, some of which are shared between CD and KT, while others are specific to the tasks of CD and KT, respectively. 3) In the Model Implementation module, we develop reusable components between CD and KT for portable model implementation. 4) In the Training Control module, from the perspective of training methodologies (such as general training, adversarial training, and other methods), we develop various training templates that can be utilized by both CD and KT models. 5) In the Model Evaluation module, we design distinct evaluation templates based on different assessment types, some of which are shared between CD and KT (e.g., the PredictionEvalTPL), while

others are specific to CD (e.g., the IdentifiabilityEvalTPL).

For the challenge of unified management of multifaceted data, we devise a series of protocols for data processing (detailed in Section 4.2) to manage data efficiently. For the challenge of compatibility for existing task scenarios and future task scenarios, the modularized and templated design can support the challenge in a user-friendly manner. When we face new task scenarios, what we need to consider is to follow relevant protocols to develop new templates to support new models (detailed in Section 5.2).

5 Usage of EduStudio

The code example of running a model is illustrated in Fig. 6. The function `run_edustudio` is the entry point for the whole experimental process including running an existing model and running a customized model.

5.1 Running existing models

To run an existing model, we need to specify at least the dataset name (i.e., dataset parameter in `run_edustudio`) and template name in each step in the algorithm workflow (i.e., the `cls` or `clses` key in corresponding parameter dictionary). The corresponding templates of models are detailed in online Reference Table. In addition, users could also specify some parameters in the parameter dictionary to replace the lower-priority configuration. For instance, the `emb_size` parameter in `modeltpl_cfg_dict` would replace the default configuration of `MyModel`.

5.2 Implementing new templates

We can implement a new template by inheriting an existing template (i.e., a Python class). To run a customized model or replace an existing template, we just need to specify the address of corresponding template class as the value of `cls` or `clses` key instead of the value of string type. In Fig. 6, the training template and model template are customized. We can specify the class object in `cls` key to implement customization instead of the template name. It can be seen that EduStudio is highly flexible and can cover the new things that appear at each step. In response to how to implement a new template, we have placed this part of the content in the **developer guide** of the explanatory document, which can help developers quickly develop custom templates.

6 Eco-services of EduStudio

To further enable more researchers to understand and quickly

Table 4 Path Management in EduStudio. We normalize the user’s working directory

Directory	Note
<code><project>/data/<dataset>/rawdata/</code>	Store the raw data files of dataset.
<code><project>/data/<dataset>/mldata/</code>	Store data files in a standardized format.
<code><project>/data/<dataset>/cachedata/</code>	Store data files in a format that is convenient for model usage.
<code><project>/conf/<dataset>/</code>	Store configuration files in YAML format.
<code><project>/archive/<dataset>/<TrainTPL>/<ModelTPL>/<ID></code>	Store logs of completed experiments.
<code><project>/temp/<dataset>/<TrainTPL>/<ModelTPL>/<ID></code>	Store logs of ongoing or failed experiments.

¹⁾ The symbol `<>` denotes the placeholder: 1) `<project>`: project work directory; 2) `<dataset>`: dataset name; 3) `<ID>`: identifier of one experiment; 4) `<TrainTPL>`: training template class name; 5) `<ModelTPL>`: model template class name.

```
from edustudio.quickstart import run_edustudio
from edustudio.model import GDBaseModel
from edustudio.traintpl import GDBaseTrainTPL

class MyModel(GDBaseModel): # inherit basic model template
    default_cfg = {'emb_size': 64, 'lambda': 1.0}
    ...

class MyTrainTPL(GDBaseTrainTPL): # inherit basic training template
    default_cfg = {'batch_size': 64}
    ...

run_edustudio(
    dataset='assist-0910', # specify dataset name
    cfg_file_name=None, # specify configuration filename
    traintpl_cfg_dict={
        'cls': MyTrainTPL, # specify customized training template
        'batch_size': 128 # specify batch size of MyTrainTPL
    },
    datatpl_cfg_dict={
        'cls': 'CDInterDataTPLExtends' # specify name of data template
    },
    modeltpl_cfg_dict={
        'cls': MyModel, # specify customized model
        'emb_size': 32 # specify embedding size of MyModel
    },
    evaltpl_cfg_dict={
        # specify evaluation templates
        'cls': ['PredictionEvalTPL', 'InterpretabilityEvalTPL'],
        # specify used metrics in InterpretabilityEvalTPL
        'InterpretabilityEvalTPL': {'use_metrics': ['auc', 'acc']}
    },
    frame_cfg_dict={
        'CFG_FOLDER_PATH': '/home/edustudio/conf', # folder path of config file
    }
)
```

Fig. 6 Code example of EduStudio usage

participate in the field of student cognitive modeling, we offer some eco-services including a Github repository and Leaderboard website surrounding EduStudio.

6.1 Awesome-student-cognitive-modeling repository

The Github repository *awesome-student-cognitive-modeling* collects valuable resources about student cognitive modeling:

- Dataset collection and description. Here, we collect available public datasets for educational data mining and provide a detailed description for each dataset. We summarize the characteristics of each dataset to facilitate researchers in efficient selection of the dataset that is applicable to their current research.
- Research direction categorization. We summarize existing research directions in student cognitive modeling including detailed description, representative papers, and commonly used datasets of each research direction. This enables researchers to swiftly comprehend the student cognitive modeling.
- Paper collection and categorization. We collect and keep up-to-date with the latest related literature. The collected papers can be categorized into: 1) research papers; 2) survey papers; 3) dataset papers. For research papers, we also make a detailed categorization. We illustrate data usage, technique usage, and research direction of each paper, which facilitates researchers to rapidly grasp the content of these papers.

6.2 Leaderboard

To ensure the reproducibility and comparison of various student cognitive modeling models, we provide a public leaderboard. As illustrated in Fig. 7, there are two major features: Task Selection and Detailed Leaderboard. The former requires users to specify elements such as task type and dataset. The latter provides a comprehensive comparison between models in the form of graphs and tables based on specified elements.

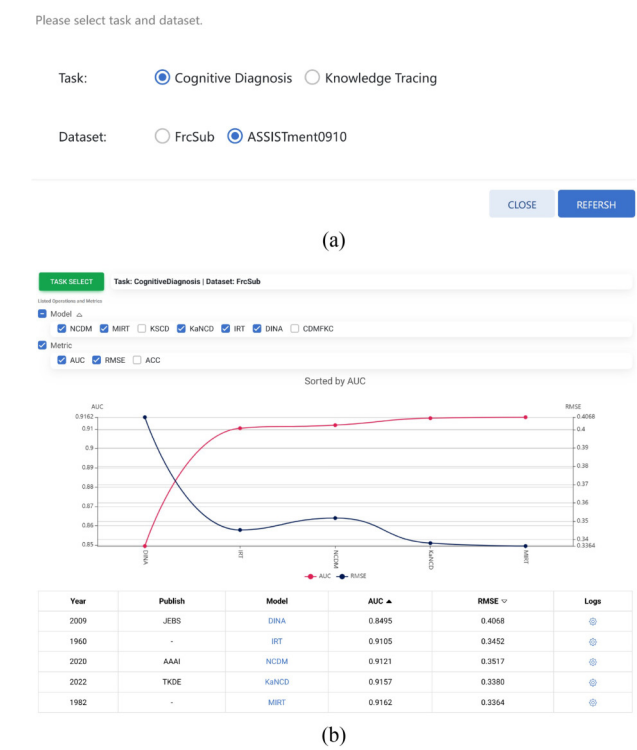


Fig. 7 Frontend of leaderboard. (a) Task selection; (b) detailed leaderboard

To support all users in uploading their experiment results, we provide a portable processing flow. In EduStudio, each experiment eventually forms a specific log directory (as depicted in Table 4). After users submit their own experiment log directory to the specific github repository, the Python script could process the new experiments and convert them into .json files required by Leaderboard frontend, and the Leaderboard frontend will automatically display the new experimental results according to the json files.

7 Comparison with existing libraries

With the growing attention from researchers toward student cognitive modeling, in the past few years, there has been a successive release of open-source algorithm libraries. Like existing libraries, EduStudio is also built using PyTorch. We summarize and compare the characteristics of existing student cognitive modeling libraries in Table 5.

EduStudio boasts a more extensive collection of models compared to the existing libraries, thereby reducing the burden of extensive model reimplementaion. Specifically, when considering individual tasks such as CD or KT, the number of models in EduStudio also surpasses those in the existing libraries. Regarding the support for datasets, our EduStudio supports a greater number of datasets. Furthermore, we provide a comprehensive data preparation process, tailored a data status protocol, middle data format protocol, and atomic data operation protocol for the data.

EduStudio supports more features, including 1) From the perspective of student cognitive modeling, integrating CD and KT, rather than considering individual tasks alone, which not only facilitates communication among researchers from both communities but also encourages the integration of the two

Table 5 Comparison with existing libraries

Library	#CD Models	#KT Models	#Datasets	Modularization	Templatization	Eco-services	Release year
EduCDM [26]	9	0	0	Low	No	Datasets	2021
EduKTM [27]	0	9	0	Low	No	Datasets	2021
pyKT [28]	0	27	13	Low	No	No	2022
EduStudio	16	29	18	High	Yes	Datasets, papers Journals & conferences Leaderboard	2023

types of student cognitive modeling approaches into one unified model. 2) The modularized and templated design makes the library highly reusable and flexible. 3) Providing comprehensive eco-services encourages more researchers to understand and participate in this field.

8 Future directions

In this section, we first discuss the research trend of student cognitive modeling. Subsequently, we talk about future work of EduStudio based on the research trend and existing limitations of EduStudio.

8.1 Research trend of student cognitive modeling

For the research trend of student cognitive modeling, we summarize some aspects according to current hotspots and opportunities.

- **Data perspective.** From data perspective, multimodal and cold-start research are two promising directions. Multimodal student cognitive modeling [88,89] aims to employ multi-modal data from student-side, exercise-side, and KC-side. Existing related work covers studies related to cold-start students [90–94], cold-start exercises [95], and cold-start KCs [96].
- **Model perspective.** From model perspective, student cognitive modeling with Large Language Models (LLMs) is emerging as a mainstream trend. LLMs have recently attracted global attention in various fields, leading some researchers to incorporate relevant technologies into student cognitive modeling [94,97–100].
- **Evaluation perspective.** From evaluation perspective, beyond accuracy evaluation, recently more researchers propose various evaluation aspects based on students’ cognitive characteristics. Fairness has consistently been a trending topic in the trustworthy AI [101–103], and ensuring fairness in education is also essential. Recently, an increasing number of researchers are delving into the fair student cognitive modeling [44,45,104–107]. IC-IDM [85] propose the identifiability evaluation, which aims to measure the discrepancy between cognitive ability of students with the same response distribution.

8.2 Future work of EduStudio

Here, we discuss the future work for EduStudio based on existing limitations and the research trend of student cognitive modeling.

- **Implement models including more scenarios.** EduStudio adopts a modular and template-based design, focusing

on balancing commonality and diversity, but it lacks sufficient consideration for diverse scenarios of student cognitive modeling. The model integration for specific scenarios is not comprehensive (such as the cold-start [90–94] and causality-based [104,108]). As described in the research trend, we will see the emergence of more new scenarios. Therefore, we will keep track of the developments in the field of student cognitive modeling and promptly implement relevant models.

- **Integrate models of downstream educational applications.** Student cognitive modeling has a series of downstream applications, among which the two most representative types are educational recommendation systems and Computerized Adaptive Testing (CAT). Educational recommendation systems aim to recommend relevant learning resources for students, such as learning path recommendation [8–10,109], course recommendation [110,111], and exercise recommendation [11,112,113]. CAT aims to provide tests that adapt dynamically to each student by tailoring test exercises based on the student’s performance [114]. The CD model is an essential component of CAT, as CAT requires CD to continuously assess students’ cognitive states [115,116]. In the future, we may consider integrating models of downstream applications based on student cognitive modeling.
- **Refine and update the eco-services promptly.** The current eco-services still requires refinement, such as enhancing the comprehensiveness and richness of the awesome-student-cognitive-modeling repository. As the trends of the future continue to change, we will update the latest content into the eco-services to ensure that it remains up-to-date and continually improve the usage of EduStudio.

9 Conclusions

In this paper, we released a unified library EduStudio for student cognitive modeling. Compared to existing libraries, we unified cognitive diagnosis and knowledge tracing, which not only enable the reusability within individual category but also facilitate sufficient reusability between them. In addition, our EduStudio is modularized and templated design when implementing models, which sufficiently improves reusability and flexibility. To further enable more researchers to understand and quickly participate in the field of student cognitive modeling, we also offered a range of user-friendly eco-services surrounding EduStudio.

Acknowledgements This work was supported in part by grants from the National Science and Technology Major Project, China (Grant No.

2021ZD0111802), the National Natural Science Foundation of China (Grant Nos. 72188101, 62406096, and 62376086), and the Fundamental Research Funds for the Central Universities, China (Grant No. JZ2024HGQB0093).

Competing interests The authors declare that they have no competing interests or financial conflicts to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Roy A, Kim S, Christensen C, Cincebeaux M. Detecting educational content in online videos by combining multimodal cues. In: Proceedings of the NeurIPS' 2023 Workshop on Generative AI for Education. 2023
- Ait Khayi N. Deep knowledge tracing using temporal convolutional networks. In: Proceedings of the Workshop Artificial Intelligence for Education. 2021
- Caines A, Benedetto L, Taslimipoor S, Davis C, Gao Y, Andersen Ø E, Yuan Z, Elliott M, Moore R, Bryant C, Rei M, Yannakoudakis H, Mullooly A, Nicholls D, Buttery P. On the application of large language models for language teaching and assessment technology. In: Proceedings of the Workshop on Empowering Education with LLMs – the Next-Gen Interface and Content Generation 2023 Co-Located with 24th International Conference on Artificial Intelligence in Education. 2023
- Deek F P, Hiltz S R, Kimmel H, Rotter N. Cognitive assessment of students' problem solving and program development skills. *Journal of Engineering Education*, 1999, 88(3): 317–326
- Huang L V, Bardos A N, D'Amato R C. Identifying students with learning disabilities: composite profile analysis using the cognitive assessment system. *Journal of Psychoeducational Assessment*, 2010, 28(1): 19–30
- Widada W. Profile of cognitive structure of students in understanding the concept of real analysis. *Infinity Journal*, 2016, 5(2): 83–98
- Liu F, Hu X, Liu S, Bu C, Wu L. Meta multi-agent exercise recommendation: a game application perspective. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 1441–1452
- Jiang B, Li X, Yang S, Kong Y, Cheng W, Hao C, Lin Q. Data-driven personalized learning path planning based on cognitive diagnostic assessments in MOOCs. *Applied Sciences*, 2022, 12(8): 3982
- Lou P. Learning path recommendation of intelligent education based on cognitive diagnosis. *International Journal of Emerging Technologies in Learning*, 2023, 18(13): 104–119
- Cai D, Zhang Y, Dai B. Learning path recommendation based on knowledge tracing model and reinforcement learning. In: Proceedings of the 5th IEEE International Conference on Computer and Communications. 2019, 1881–1885
- Ai F, Chen Y, Guo Y, Zhao Y, Wang Z, Fu G, Wang G. Concept-aware deep knowledge tracing and exercise recommendation in an online learning system. In: Proceedings of the 12th International Educational Data Mining Society. 2019
- Lian D, Wu Y, Ge Y, Xie X, Chen E. Geography-aware sequential location recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020, 2009–2019
- Wang H, Xu T, Liu Q, Lian D, Chen E, Du D, Wu H, Su W. MCNE: an end-to-end framework for learning multiple conditional network representations of social network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2019, 1064–1072
- Gao C, Wang S, Li S, Chen J, He X, Lei W, Li B, Zhang Y, Jiang P. CIRS: bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems*, 2024, 42(1): 14
- Zhuang Y, Liu Q, Zhao G, Huang Z, Huang W, Pardos Z A, Chen E, Wu J, Li X. A bounded ability estimation for computerized adaptive testing. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2024, 111
- Liu Q. Towards a new generation of cognitive diagnosis. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. 2021, 4961–4964
- Liu Y, Zhang T, Wang X, Yu G, Li T. New development of cognitive diagnosis models. *Frontiers of Computer Science*, 2023, 17(1): 171604
- Abdelrahman G, Wang Q, Nunes B. Knowledge tracing: a survey. *ACM Computing Surveys*, 2023, 55(11): 224
- Shen S, Liu Q, Huang Z, Zheng Y, Yin M, Wang M, Chen E. A survey of knowledge tracing: models, variants, and applications. *IEEE Transactions on Learning Technologies*, 2024, 17: 1898–1919
- Liu H, Zhang T, Li F, Yu M, Yu G. A probabilistic generative model for tracking multi-knowledge concept mastery probability. *Frontiers of Computer Science*, 2024, 18(3): 183602
- Gurria A. PISA 2015 results in focus. Paris: Organisation for Economic Co-operation and Development (OECD), 2016
- Schleicher A. PISA 2018: insights and interpretations. Paris: Organisation for Economic Co-operation and Development (OECD), 2019
- İdil Ş, Gülen S, Dönmez İ. What should we understand from PISA 2022 results? *Journal of STEAM Education*, 2024, 7(1): 1–9
- Piech C, Bassen J, Huang J, Ganguli S, Sahami M, Guibas L, Sohl-Dickstein J. Deep knowledge tracing. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. 2015, 505–513
- Huang Z, Liu Q, Zhai C, Yin Y, Chen E, Gao W, Hu G. Exploring multi-objective exercise recommendations in online education systems. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019, 1261–1270
- BigData-USTC. EduCDM. See [Github.com/bigdataustc/EduCDM](https://github.com/bigdataustc/EduCDM) website, 2021
- BigData-USTC. EduKTM. See [Github.com/bigdataustc/EduKTM](https://github.com/bigdataustc/EduKTM) website, 2021
- Liu Z, Liu Q, Chen J, Huang S, Tang J, Luo W. pyKT: a Python library to benchmark deep learning based knowledge tracing models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 18542–18555
- Nagatani K, Zhang Q, Sato M, Chen Y Y, Chen F, Ohkuma T. Augmenting knowledge tracing by considering forgetting behavior. In: Proceedings of the World Wide Web Conference 2019. 2019, 3101–3107
- Shen S, Liu Q, Chen E, Huang Z, Huang W, Yin Y, Su Y, Wang S. Learning process-consistent knowledge tracing. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021, 1452–1460

31. Shen S, Chen E, Liu Q, Huang Z, Huang W, Yin Y, Su Y, Wang S. Monitoring student progress for learning process-consistent knowledge tracing. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35(8): 8213–8227
32. de la Torre J. Dina model and parameter estimation: a didactic. *Journal of Educational and Behavioral Statistics*, 2009, 34(1): 115–130
33. Wang F, Huang Z, Liu Q, Chen E, Yin Y, Ma J, Wang S. Dynamic cognitive diagnosis: an educational priors-enhanced deep knowledge tracing perspective. *IEEE Transactions on Learning Technologies*, 2023, 16(3): 306–323
34. Rasch G. *Probabilistic Models for Some Intelligence and Attainment Tests*. Chicago: University of Chicago Press, 1980
35. DeVellis R F. *Classical test theory*. *Medical Care*, 2006, 44(11): S50–S59
36. Junker B W, Sijtsma K. Cognitive assessment models with few assumptions, and connections with nonparametric item response theory. *Applied Psychological Measurement*, 2001, 25(3): 258–272
37. Leighton J P, Gierl M J, Hunka S M. The attribute hierarchy method for cognitive assessment: a variation on tatsuoaka's rule-space approach. *Journal of Educational Measurement*, 2004, 41(3): 205–237
38. Wang F, Liu Q, Chen E, Huang Z, Chen Y, Yin Y, Huang Z, Wang S. Neural cognitive diagnosis for intelligent education systems. In: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. 2020, 6153–6161
39. Wang F, Liu Q, Chen E, Huang Z, Yin Y, Wang S, Su Y. NeuralCD: a general framework for cognitive diagnosis. *IEEE Transactions on Knowledge and Data Engineering*, 2023, 35(8): 8312–8327
40. Wang X, Huang C, Cai J, Chen L. Using knowledge concept aggregation towards accurate cognitive diagnosis. In: *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2021, 2010–2019
41. Ma H, Li M, Wu L, Zhang H, Cao Y, Zhang X, Zhao X. Knowledge-sensed cognitive diagnosis for intelligent education platforms. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, 1451–1460
42. Li S, Guan Q, Fang L, Xiao F, He Z, He Y, Luo W. Cognitive diagnosis focusing on knowledge concepts. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, 3272–3281
43. Zhou Y, Liu Q, Wu J, Wang F, Huang Z, Tong W, Xiong H, Chen E, Ma J. Modeling context-aware features for cognitive diagnosis in student learning. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, 2420–2428
44. Zhang Z, Wu L, Liu Q, Liu J, Huang Z, Yin Y, Zhuang Y, Gao W, Chen E. Understanding and improving fairness in cognitive diagnosis. *Science China Information Sciences*, 2024, 67(5): 152106
45. Zhang Z, Liu Q, Jiang H, Wang F, Zhuang Y, Wu L, Gao W, Chen E. FairLISA: fair user modeling with limited sensitive attributes information. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 2023, 41432–41450
46. Huang J, Liu Q, Wang F, Huang Z, Fang S, Wu R, Chen E, Su Y, Wang S. Group-level cognitive diagnosis: a multi-task learning perspective. In: *Proceedings of 2021 IEEE International Conference on Data Mining*. 2021, 210–219
47. Gao W, Liu Q, Huang Z, Yin Y, Bi H, Wang M C, Ma J, Wang S, Su Y. RCD: relation map driven cognitive diagnosis for intelligent education systems. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2021, 501–510
48. Li J, Wang F, Liu Q, Zhu M, Huang W, Huang Z, Chen E, Su Y, Wang S. HierCDF: a Bayesian network-based hierarchical cognitive diagnosis framework. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, 904–913
49. Chen X, Wu L, Liu F, Chen L, Zhang K, Hong R, Wang M. Disentangling cognitive diagnosis with limited exercise labels. In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. 2024, 792
50. Xu B, Huang Z, Liu J, Shen S, Liu Q, Chen E, Wu J, Wang S. Learning behavior-oriented knowledge tracing. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2023, 2789–2800
51. Corbett A T, Anderson J R. Knowledge tracing: modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 1994, 4(4): 253–278
52. Käser T, Klingler S, Schwing A G, Gross M. Dynamic Bayesian networks for student modeling. *IEEE Transactions on Learning Technologies*, 2017, 10(4): 450–462
53. Cen H, Koedinger K, Junker B. Learning factors analysis – a general method for cognitive model evaluation and improvement. In: *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*. 2006, 164–175
54. Pavlik P I, Cen H, Koedinger K R. Performance factors analysis – a new alternative to knowledge tracing. In: *Proceedings of the 2009 Conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*. 2009, 531–538
55. Vie J J, Kashima H. Knowledge tracing machines: factorization machines for knowledge tracing. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. 2019, 750–757
56. Yeung C K, Yeung D Y. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In: *Proceedings of the 5th Annual ACM Conference on Learning at Scale*. 2018, 5
57. Minn S, Yu Y, Desmarais M C, Zhu F, Vie J J. Deep knowledge tracing and dynamic student classification for knowledge tracing. In: *Proceedings of 2018 IEEE International Conference on Data Mining*. 2018, 1182–1187
58. Sonkar S, Lan A S, Waters A E, Grimaldi P, Baraniuk R G. qDKT: question-centric deep knowledge tracing. In: *Proceedings of the 13th International Conference on Educational Data Mining*. 2020
59. Zhang J, Shi X, King I, Yeung D Y. Dynamic key-value memory networks for knowledge tracing. In: *Proceedings of the 26th International Conference on World Wide Web*. 2017, 765–774
60. Abdelrahman G, Wang Q. Knowledge tracing with sequential key-value memory networks. In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2019, 175–184
61. Liu Q, Huang Z, Yin Y, Chen E, Xiong H, Su Y, Hu G. EKT: exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2021, 33(1): 100–115
62. Pandey S, Karypis G. A self attentive model for knowledge tracing. In: *Proceedings of the 12th International Conference on Educational Data Mining*. 2019, 384–389
63. Choi Y, Lee Y, Cho J, Baek J, Kim B, Cha Y, Shin D, Bae C, Heo J. Towards an appropriate query, key, and value computation for knowledge tracing. In: *Proceedings of the 7th ACM Conference on Learning@Scale*. 2020, 341–344
64. Shin D, Shim Y, Yu H, Lee S, Kim B, Choi Y. SAINT+: Integrating temporal features for EdNet correctness prediction. In: *Proceedings of the 11th International Learning Analytics and Knowledge Conference*. 2021, 490–496
65. Nakagawa H, Iwasawa Y, Matsuo Y. Graph-based knowledge tracing:

- modeling student proficiency using graph neural network. In: Proceedings of 2019 IEEE/WIC/ACM International Conference on Web Intelligence. 2019, 156–163
66. Yang Y, Shen J, Qu Y, Liu Y, Wang K, Zhu Y, Zhang W, Yu Y. GIKT: a graph-based interaction model for knowledge tracing. In: Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference. 2020, 299–315
67. Huang Z, Liu Q, Chen Y, Wu L, Xiao K, Chen E, Ma H, Hu G. Learning or forgetting? A dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems*, 2020, 38(2): 19
68. Yeung C K. Deep-IRT: make deep learning based knowledge tracing explainable using item response theory. In: Proceedings of the 12th International Conference on Educational Data Mining. 2019
69. Gan W, Sun Y, Sun Y. Knowledge interaction enhanced sequential modeling for interpretable learner knowledge diagnosis in intelligent tutoring systems. *Neurocomputing*, 2022, 488: 36–53
70. McKinley R L, Reckase M D. The use of the general rasch model with multidimensional item response data. Iowa City: American College Testing, 1982
71. Su Y, Liu Q, Liu Q, Huang Z, Yin Y, Chen E, Ding C, Wei S, Hu G. Exercise-enhanced sequential modeling for student performance prediction. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence. 2018
72. Lee J, Yeung D Y. Knowledge query network for knowledge tracing: how knowledge interacts with skills. In: Proceedings of the 9th International Conference on Learning Analytics & Knowledge. 2019, 491–500
73. Ghosh A, Heffernan N, Lan A S. Context-aware attentive knowledge tracing. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020, 2330–2339
74. Shen S, Liu Q, Chen E, Wu H, Huang Z, Zhao W, Su Y, Ma H, Wang S. Convolutional knowledge tracing: modeling individualization in student learning process. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020, 1857–1860
75. Pandey S, Srivastava J. RKT: relation-aware self-attention for knowledge tracing. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. 2020, 1205–1214
76. Guo X, Huang Z, Gao J, Shang M, Shu M, Sun J. Enhancing knowledge tracing via adversarial training. In: Proceedings of the 29th ACM International Conference on Multimedia. 2021, 367–375
77. Long T, Liu Y, Shen J, Zhang W, Yu Y. Tracing knowledge state with individual cognition and acquisition estimation. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021, 173–182
78. Wang C, Ma W, Zhang M, Lv C, Wan F, Lin H, Tang T, Liu Y, Ma S. Temporal cross-effects in knowledge tracing. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining. 2021, 517–525
79. Ma H, Wang J, Zhu H, Xia X, Zhang H, Zhang X, Zhang L. Reconciling cognitive modeling with knowledge forgetting: a continuous time-aware neural network approach. In: Proceedings of the 31st International Joint Conference on Artificial Intelligence. 2022, 2174–2181
80. Lee W, Chun J, Lee Y, Park K, Park S. Contrastive learning for knowledge tracing. In: Proceedings of the ACM Web Conference 2022. 2022, 2330–2338
81. Shen S, Huang Z, Liu Q, Su Y, Wang S, Chen E. Assessing student's dynamic knowledge state by exploring the question difficulty effect. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2022, 427–437
82. Chen J, Liu Z, Huang S, Liu Q, Luo W. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In: Proceedings of the 37th AAAI Conference on Artificial Intelligence. 2023, 14196–14204
83. Liu Z, Liu Q, Chen J, Huang S, Luo W. simpleKT: a simple but tough-to-beat baseline for knowledge tracing. In: Proceedings of the 11th International Conference on Learning Representations. 2023
84. Yin Y, Dai L, Huang Z, Shen S, Wang F, Liu Q, Chen E, Li X. Tracing knowledge instead of patterns: stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023. 2023, 855–864
85. Li J, Liu Q, Wang F, Liu J, Huang Z, Yao F, Zhu L, Su Y. Towards the identifiability and explainability for personalized learner modeling: an inductive paradigm. In: Proceedings of the ACM Web Conference 2024. 2024, 3420–3431
86. Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R. Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. 2012, 214–226
87. Hardt M, Price E, Srebro N. Equality of opportunity in supervised learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. 2016, 3323–3331
88. Yu J, Wang Y, Zhong Q, Luo G, Mao Y, Sun K, Feng W, Xu W, Cao S, Zeng K, Yao Z, Hou L, Lin Y, Li P, Zhou J, Xu B, Li J, Tang J, Sun M. MOOCubeX: a large knowledge-centered repository for adaptive learning in MOOCs. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. 2021, 4643–4652
89. Song L, He M, Shang X, Yang C, Liu J, Yu M, Lu Y. A deep cross-modal neural cognitive diagnosis framework for modeling student performance. *Expert Systems with Applications*, 2023, 230: 120675
90. Zhao J, Bhatt S, Thille C, Gattani N, Zimmaro D. Cold start knowledge tracing with attentive neural turing machine. In: Proceedings of the 7th ACM Conference on Learning @ Scale. 2020, 333–336
91. Gao W, Wang H, Liu Q, Wang F, Lin X, Yue L, Zhang Z, Lv R, Wang S. Leveraging transferable knowledge concept graph embedding for cold-start cognitive diagnosis. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2023, 983–992
92. Liu S, Shen J, Qian H, Zhou A. Inductive cognitive diagnosis for fast student learning in web-based intelligent education systems. In: Proceedings of the ACM Web Conference 2024. 2024, 4260–4271
93. Gao W, Liu Q, Wang H, Yue L, Bi H, Gu Y, Yao F, Zhang Z, Li X, He Y. Zero-1-to-3: domain-level zero-shot cognitive diagnosis via one batch of early-bird students towards three diagnostic objectives. In: Proceedings of the 38th AAAI Conference on Artificial Intelligence. 2024, 8417–8426
94. Jung H, Yoo J, Yoon Y, Jang Y. CLST: cold-start mitigation in knowledge tracing by aligning a generative language model as a students' knowledge tracer. 2024, arXiv preprint arXiv: 2406.10296
95. Liu G, Zhan H, Kim J J. Question difficulty consistent knowledge tracing. In: Proceedings of the ACM Web Conference 2024. 2024, 4239–4248
96. Das R, Zhang J, Baker R S, Scruggs R. A new interpretation of knowledge tracing models' predictive performance in terms of the cold start problem. In: Proceedings of the 14th International Conference on Educational Data Mining. 2021
97. Zhang H, Liu Z, Huang S, Shang C, Zhan B, Jiang Y. Improving low-resource knowledge tracing tasks by supervised pre-training and importance mechanism fine-tuning. 2024, arXiv preprint arXiv:

2403.06725

98. Zhan B, Guo T, Li X, Hou M, Liang Q, Gao B, Luo W, Liu Z. Knowledge tracing as language processing: a large-scale autoregressive paradigm. In: Proceedings of the 25th International Conference on Artificial Intelligence in Education. 2024, 177–191
99. Fu L, Guan H, Du K, Lin J, Xia W, Zhang W, Tang R, Wang Y, Yu Y. SINKT: a structure-aware inductive knowledge tracing model with large language model. 2024, arXiv preprint arXiv: 2407.01245
100. Lee U, Bae J, Kim D, Lee S, Park J, Ahn T, Lee G, Stratton D, Kim H. Language model can do knowledge tracing: simple but effective method to integrate language model and knowledge tracing task. 2024, arXiv preprint arXiv: 2406.02893
101. Caton S, Haas C. Fairness in machine learning: a survey. *ACM Computing Surveys*, 2024, 56(7): 166
102. Shao P, Wu L, Zhang K, Lian D, Hong R, Li Y, Wang M. Average user-side counterfactual fairness for collaborative filtering. *ACM Transactions on Information Systems*, 2024, 42(5): 140
103. Chen L, Wu L, Zhang K, Hong R, Lian D, Zhang Z, Zhou J, Wang M. Improving recommendation fairness via data augmentation. In: Proceedings of the ACM Web Conference 2023. 2023, 1012–1020
104. Zhang D, Zhang K, Wu L, Tian M, Hong R, Wang M. Path-specific causal reasoning for fairness-aware cognitive diagnosis. In: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2024
105. Tschitschek S, Knobelsdorf M, Singla A. Equity and fairness of Bayesian knowledge tracing. In: Proceedings of the 15th International Conference on Educational Data Mining. 2022
106. Barrett J, Day A, Gal K. Improving model fairness with time-augmented Bayesian knowledge tracing. In: Proceedings of the 14th Learning Analytics and Knowledge Conference. 2024, 46–54
107. Doroudi S, Brunskill E. Fairer but not fair enough on the equitability of knowledge tracing. In: Proceedings of the 9th International Conference on Learning Analytics & Knowledge. 2019, 335–339
108. Wu C, Wang X, Lian D, Xie X, Chen E. A causality inspired framework for model interpretation. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023, 2731–2741
109. Wan H, Che B, Luo H, Luo X. Learning path recommendation based on knowledge tracing and reinforcement learning. In: Proceedings of 2023 IEEE International Conference on Advanced Learning Technologies. 2023, 55–57
110. Tian X, Liu F. Capacity tracing-enhanced course recommendation in MOOCs. *IEEE Transactions on Learning Technologies*, 2021, 14(3): 313–321
111. Ban Q, Wu W, Hu W, Lin H, Zheng W, He L. Knowledge-enhanced multi-task learning for course recommendation. In: Proceedings of the 27th International Conference on Database Systems for Advanced Applications. 2022, 85–101
112. Ma H, Huang Z, Tang W, Zhang X. Exercise recommendation based on cognitive diagnosis and neutrosophic set. In: Proceedings of the 25th IEEE International Conference on Computer Supported Cooperative Work in Design. 2022, 1467–1472
113. Cheng Y, Li M, Chen H, Cai Y, Sun H, Zou H, Zhang G. Exercise recommendation method combining NeuralCD and NeuMF models. In: Proceedings of the 7th Annual International Conference on Network and Information Systems for Computers. 2021, 646–651
114. Wang F, Gao W, Liu Q, Li J, Zhao G, Zhang Z, Huang Z, Zhu M, Wang S, Tong W, Chen E. A survey of models for cognitive diagnosis: new developments and future directions. 2024, arXiv preprint arXiv: 2407.05458
115. Yu J, Zhuang Y, Huang Z, Liu Q, Li X, Li R, Chen E. A unified adaptive testing system enabled by hierarchical structure search. In:

Proceedings of the 41st International Conference on Machine Learning. 2024

116. Bi H, Liu Q, Wu H, He W, Huang Z, Yin Y, Ma H, Su Y, Wang S, Chen E. Model-agnostic adaptive testing for intelligent education systems via meta-learned gradient embeddings. *ACM Transactions on Intelligent Systems and Technology*, 2024



Le Wu is currently a professor at the Hefei University of Technology (HFUT), China. She received her PhD degree from the University of Science and Technology of China (USTC), China. Her general area of research interests are data mining, recommender systems, and responsible user modeling. She has published more than 60 papers in referred journals and conferences, such as IEEE TKDE, NIPS, SIGIR, WWW, and AAAI. Dr. Le Wu is the recipient of the Best of SDM 2015 Award, and the Distinguished Dissertation Award from the China Association for Artificial Intelligence (CAAI) 2017.



Xiangzhi Chen is currently pursuing a PhD degree at Hefei University of Technology, China. He received the BE degree from Hefei University of Technology, China in 2021. His research interest lies on educational data mining and artificial intelligence for education. He has published articles in international conferences and journals, such as NeurIPS and IEEE TKDE.



Fei Liu received her PhD degree at School of Computer Science and Information Engineering in Hefei University of Technology, China in 2023. She is currently a postdoctoral fellow at Hefei University of Technology, China. Her research mainly lies in educational data mining. She has published articles in international conferences and journals, such as SIGKDD, NeurIPS, ACM Transactions on Information Systems (ACM TOIS), IEEE Transactions on Fuzzy Systems (IEEE TFS), IEEE Transactions on Emerging Topics in Computational Intelligence (IEEE TETCI), and Information Fusion.



Junsong Xie is currently pursuing a PhD degree at Hefei University of Technology (HFUT), China. He received the master's degree from University of Science and Technology of China (USTC), China. His major research interest lies on data mining and recommender systems.



Chenao Xia is currently pursuing the MS degree with the School of Computer Science and Technology, Hefei University of Technology, China. He received the BE degree from Anhui Normal University, China in 2022. His current research interests include Educational Data Mining and Deep Learning.



Zhengtao Tan is currently working towards a Master's degree at Hefei University of Technology, China. He received his undergraduate degree from Hefei University of Technology, China in 2022. His research interests include cognitive diagnosis and invariant learning.



Mi Tian is currently pursuing a graduate degree at Hefei University of Technology, China. She completed her undergraduate studies at Shanxi University of Finance and Economics, China. Her primary research interest lies in the field of Educational Data Mining, such as Cognitive Diagnosis and Computerized Adaptive Testing.



Jinglong Li is currently pursuing a master's degree at Hefei University of Technology, China. He received his bachelor's degree from South-Central Minzu University, China in 2023. His major research interest lies in education data mining and out-of-distribution generalization.



Kun Zhang received a PhD degree in computer science and technology from the University of Science and Technology of China in 2019. He is currently a faculty member at the Hefei University of Technology (HFUT), China. His research interests include Natural Language Understanding and Recommender Systems. He has published several papers in refereed journals and conferences, such as the IEEE TSMC:S, IEEE TKDE, ACM TKDD, AAAI, KDD, ACL, and ICDM. He received the KDD 2018 Best Student Paper Award.



Defu Lian received the PhD degree in computer science from the University of Science and Technology of China (USTC), China in 2014. He is currently a professor of the School of Computer Science and Technology, USTC. He has published prolifically in referred journals and conference proceedings, such as ACM Transactions on Information Systems and IEEE Transaction on Knowledge and Data

Engineering, IEEE International Conference on Data Mining, ACM SIGKDD Conference on Knowledge Discovery and Data Mining, and ACM International World Wide Web Conferences. His current research interest includes spatial data mining, recommender systems, and learning to hash.



Richang Hong is currently a professor at Hefei University of Technology, China. He received the PhD degree from the University of Science and Technology of China, China in 2008. He has published more than 100 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. He is a member of the Association for Computing Machinery. He was a recipient of the Best Paper award in the ACM Multimedia 2010.



Meng Wang is a professor at the Hefei University of Technology, China. He received his BE degree and PhD degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), China in 2003 and 2008, respectively. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He is an associate editor of IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT), IEEE Transactions on Multimedia (IEEE TMM), and IEEE Transactions on Neural Networks and Learning Systems (IEEE TNNLS).