sustainability

MDPI

Article

Multiple Learning Features–Enhanced Knowledge Tracing Based on Learner–Resource Response Channels

Zhifeng Wang [1,*], Yulin Hou [1], Chunyan Zeng [2,*], Si Zhang [1] and Ruiqiu Ye [1]

[1] Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430079, China
[2] Hubei Key Laboratory for High-Efficiency Utilization of Solar Energy and Operation Control of Energy Storage System, Hubei University of Technology, Wuhan 430068, China
* Correspondence: zfwang@ccnu.edu.cn (Z.W.); cyzeng@hbut.edu.cn (C.Z.)

Abstract: Knowledge tracing is a crucial task that involves modeling learners' knowledge levels and predicting their future learning performance. However, traditional deep knowledge tracing approaches often overlook the intrinsic relationships among learning features, treating them equally and failing to align with real learning scenarios. To address these issues, this paper proposes the multiple learning features, enhanced knowledge tracing (MLFKT) framework. Firstly, we construct learner–resource response (LRR) channels based on psychometric theory, establishing stronger intrinsic connections among learning features and overcoming the limitations of the item response theory. Secondly, we leverage stacked auto-encoders to extract low-dimensional embeddings for different LRR channels with denser representations. Thirdly, considering the varying impact of different LRR channels on learning performance, we introduce an attention mechanism to assign distinct weights to each channel. Finally, to address the challenges of memory retention and forgetting in the learning process and to handle long-term dependency issues, we employ a bidirectional long short-term memory network to model learners' knowledge states, enabling accurate prediction of learning performance. Through extensive experiments on two real datasets, we demonstrate the effectiveness of our proposed MLFKT approach, which outperforms six traditional methods. The newly proposed method can enhance educational sustainability by improving the diagnosis of learners' self-cognitive structures and by empowering teachers to intervene and personalize their teaching accordingly.

Keywords: attention mechanisms; bidirectional long short-term memory networks; knowledge tracing; learning performance prediction

check for updates

Citation: Wang, Z.; Hou, Y.; Zeng, C.; Zhang, S.; Ye, R. Multiple Learning Features–Enhanced Knowledge Tracing Based on Learner–Resource Response Channels. *Sustainability* **2023**, *15*, 9427. https://doi.org/10.3390/su15129427

Academic Editor: Eila Jeronen

Received: 11 May 2023
Revised: 5 June 2023
Accepted: 9 June 2023
Published: 12 June 2023

1. Introduction

In the 21st century, advancements in high-tech information technologies such as artificial intelligence, big data, virtual reality, blockchain, and the Internet of things have brought disruptive changes to the field of education. These innovations have paved the way for the transformation of traditional education into a new paradigm known as smart education [1], which has greatly enhanced the sustainability of education [2]. E-learning platforms such as Coursera, Edx, and Khan Academy have gained significant popularity for sustainable education. However, the quality of learning resources provided by these platforms varies, resulting in reduced learning efficiency and quality [3]. This situation deprives learners of personalized services and prevents teachers from offering tailored learning guidance. Particularly, with the unprecedented rise of online education due to the COVID-19 pandemic, the need to provide large-scale online education with personalized learning services has become urgent and essential for sustainable education [4].

Knowledge tracing emerges as a promising solution to address this need. It leverages the abundant data generated by learners' online learning journeys to automatically track their knowledge levels, estimate their progress in acquiring specific knowledge components, and accurately predict their performance [5]. Over the past 30 years, knowledge tracing has seen significant development, and classical models such as Bayesian knowledge tracing (BKT) [6] and performance factor analysis (PFA) [7] have been extensively studied. In 2015, Piech et al. introduced recurrent neural networks (RNN) [8] into the knowledge tracing task and proposed the pioneering deep knowledge tracing (DKT) model, demonstrating its effectiveness in performance prediction and opening up new research avenues in knowledge tracing [9]. The field of knowledge tracing has since made substantial progress.

However, despite the important role of DKT in personalized e-learning services, there are still several limitations in its development. Firstly, DKT models consider features in a disordered manner. Existing deep knowledge tracing models that rely on multi-feature analysis randomly match and characterize different features, overlooking the inherent connections among them. Secondly, these models treat various learning factors equally. While existing deep knowledge tracing models consider the influence of multiple features on the learning process, they either assign equal importance to different features or lack interpretability in assessing feature significance. Thirdly, there is a lack of interpretable support from the field of education. Although deep knowledge tracing excels in characterizing learners' knowledge states and performance prediction, it treats the diagnostic process as a "black box" and fails to provide reasonable explanations grounded in educational knowledge.

To address the aforementioned limitations, this paper introduces a comprehensive approach that incorporates various learner features, resource features, and response features, all of which significantly impact the learning process. Grounded in psychometric theory and considering item response theory (IRT) [10], we argue that learners' potential knowledge states are not solely determined by their responses to specific resources. Instead, they are also influenced by tapping information associated with the resource, as well as learners' personal and cognitive attributes. Furthermore, we emphasize that responses should not be confined to simple right or wrong answers to exercises. Building upon this perspective, we propose the construction of learner–resource response (LRR) channels, which serve to expand the scope of item response theory and establish internal connections among the features. These LRR channels encompass learner features, resource features, and response features that align with specific educational assumptions. To ensure seamless integration and association of these LRR channels without introducing unnecessary model complexity, we employ stacked auto-encoders to extract low-dimensional embedding representations, resulting in denser LRR channels. Additionally, we incorporate an attention mechanism to assign distinct attention weights to individual LRR channels, enabling deep feature associations while prioritizing crucial information. To further enhance interpretability within the educational domain, we introduce the innovative use of a Bi-LSTM network [11] to model learners' knowledge states. This architecture not only excels at handling long temporal dependencies but also accommodates the dual influence of memory and forgetting factors in the educational context. Consequently, the model becomes more interpretable within the educational domain. Finally, we present a novel knowledge tracing framework that accurately models learners' knowledge and predicts their performance. By capturing changes in learners' learning states more precisely, this framework provides a comprehensive characterization of the entire learning process.

1.1. Research Questions

This paper aims to investigate multiple learning-feature-enhanced knowledge tracing and address the following specific research questions:

*RQ*1: How can we quantitatively model the intrinsic associations among learner features, resource features, and response features to effectively enhance knowledge tracing performance?

*RQ*2: How can we learn deep representations of learner features, resource features, and response features to develop more accurate models for predicting learning performance?

*RQ*3: How can we obtain interpretable learning performance prediction results to better serve educational applications in real-world learning scenarios?

1.2. Research Motivations

To achieve improved learning representations of multiple learning features and to provide answers to the research questions stated above, we have undertaken targeted efforts that distinguish our work from previous studies. The research motivations for this study are as follows:

1. Incorporating learner features, resource features, and response features based on the principles of psychometric theory, we construct LRR channels using item response theory to enhance the intrinsic associations among different learning features.
2. Utilizing stacked auto-encoders, we mine low-dimensional embedding representations of LRR channels. Moreover, by employing an attention mechanism, we assign varying weights to each LRR channel during training, allowing the model to focus more on the crucial channels. This facilitates better fusion and deeper correlation of learning features.
3. Drawing upon the classical education curve theory, we introduce Bi-LSTM into the knowledge tracing task. By considering both memory and forgetting factors, Bi-LSTM takes into account both past and future response information. This allows for comprehensive consideration of the learner's memory accumulation and forgetting degree, which are reflected in the knowledge state.

The subsequent sections of this paper are structured as follows: Section 2 provides an overview of the related work, highlighting the existing research in the field. Section 3 defines the research problems and introduces key assumptions and definitions that serve as the foundation for our proposed framework. The details of the multiple learning features, enhanced knowledge tracing (MLFKT) framework are presented in Section 4, offering comprehensive insights into its design and functionality. Section 5 presents an elaborate account of the experiments conducted. Sections 6 and 7 show a detailed analysis and discussion of the experimental results. Finally, in Section 8, we provide a summary of our work and offer an outlook on potential future directions in this domain.

2. Related Work

This section provides an overview of the current state of research on knowledge tracing and highlights the unique contributions of this paper compared to previous work.

2.1. Probabilistic Graph-Based Knowledge Tracing

The concept of knowledge tracing was initially introduced by Corbett and Anderson in 1994 within the field of intelligent education [6]. Knowledge tracing involves monitoring the evolving knowledge states of learners in practice, as well as modeling these states and predicting learners' performance in subsequent learning interactions. Over the past few decades, researchers have developed various mathematical and computational models to tackle knowledge tracing tasks [12]. These approaches encompass a wide range, from probabilistic models to deep neural networks [13]. Prior to the advent of deep learning in knowledge tracing, probabilistic graph-based methods enjoyed popularity [14].

The probabilistic graph-based model provides a framework for describing a learning task that focuses on making inferences about the distribution of unknown variables using known variables [15]. A typical example of a probabilistic graph-based knowledge tracing model is the BKT model [6], which utilizes hidden Markov models (HMM) [16] as a template-based probabilistic graph model to represent the learning process. BKT is a time-series model with an implicit layer representing the knowledge state and an observation layer representing performance data. The BKT model assumes three hypotheses: (1) learners' knowledge states are either mastered or unmastered; (2) learners do not experience forgetting during the learning process; (3) knowledge points are independent of each other. However, these hypotheses overly idealize the real learning scenarios of learners. As a result, several variants of the BKT model have been proposed to address the limitations arising from these assumptions and improve the prediction accuracy to varying degrees. Additionally, personalized learning parameters have emerged as a significant research direction in probabilistic graph-based knowledge tracing. It is believed that the prediction accuracy of the probabilistic graph-based knowledge tracing model depends on its parameters. While traditional BKT fixes four parameters in the prediction process (initial mastery probability, transition probability, slipping probability, and guessing probability), the actual learning scenario demonstrates that different learners exhibit distinct initial mastery probabilities, slipping probabilities, and guessing probabilities. Therefore, incorporating personalized learning parameters is beneficial for effectively characterizing the entire learning process of learners and improving prediction accuracy.

2.2. *Deep-Learning-Based Knowledge Tracing*

The introduction of deep learning [17–20] has brought about rapid advancements in the field of knowledge tracing [21]. In 2015, Piech et al. pioneered the use of recurrent neural networks, specifically long short-term memory (LSTM) networks [22], for knowledge tracing and proposed the DKT model, which revolutionized the representation of learners' knowledge states and the prediction of their performance [9]. Since then, an array of deep learning methods has been employed to improve the prediction of learners' knowledge states and performance [23]. Numerous advanced knowledge tracing models have emerged as a result [24–28].

DKT+ [29] introduced regularization terms for reconstruction and waviness to the loss function of the original DKT model, enhancing consistency in performance prediction. Additionally, the dynamic key–value memory network (DKVMN) [30] models the relationships between underlying concepts and directly extracts a student's mastery level of each concept. Unlike conventional memory-augmented neural networks that employ a single memory matrix or two static memory matrices, DKVMN utilizes a static key matrix to store knowledge concepts and a dynamic value matrix to store and update the mastery levels of corresponding concepts. DKVMN has expanded the new research direction of knowledge tracing by better capturing the dependency relationships between knowledge points and by addressing the issue of accurately tracking learners' mastery of specific knowledge points [31]. However, the utilization of deep learning in knowledge tracing is not limited to DKVMN, as more and more deep learning algorithms are being applied to various methodological aspects of knowledge tracing to better extract and represent the entire learning process of learners [32–34]. For instance, Wang et al. proposed a learner-personalized modeling approach based on convolutional neural networks (CNN) [35] for knowledge tracing tasks [36]. Pandey et al. introduced a method called self-attentive knowledge tracing (SAKT) for identifying relevant knowledge points from a student's past activities and for predicting mastery based on a selective subset of knowledge points [37]. Zhang et al. employed graph convolutional networks (GCN) [38] to capture higher-order information within the problem technique graph [39]. Nakagawa et al. introduced a knowledge tracing method based on graph neural networks (GNN) [40] that takes into account the potential graph structure. This approach converts the knowledge structure into a graph, enabling the transformation of the knowledge tracing task into a time series node and classification problem within the GNN framework. Various implementations of knowledge graph structures have been provided as well [41]. While deep learning-based knowledge tracing exhibits strong performance in predicting learners' performance and representing their knowledge states, its interpretability remains limited. Enhancing interpretability within the knowledge tracing domain poses a challenge for future research scholars.

2.3. *Multi-Feature Analysis in Knowledge Tracing*

With the rapid advancements in deep-learning-based knowledge tracing, researchers have recognized certain limitations in traditional deep knowledge tracing methods. These traditional approaches primarily rely on learners' answered exercises and their correctness, disregarding the abundance of interactive information present in real-world learning scenarios. Factors such as knowledge point difficulty [26], exercise text [42], learner ability [3], response duration time [43], and knowledge associations [44] are often overlooked. As a result, scholars have endeavored to integrate additional edge features into the traditional deep knowledge tracing model to enhance its effectiveness [26,45–47]. Liang et al. [48] introduced numerous edge features through extensive feature engineering, randomly matching and characterizing these features using artificial neural networks, and effectively enhancing the prediction accuracy of deep knowledge tracing. Cheung et al. [49] developed an automated system to uniformly process edge features and embed them into the deep knowledge tracing model, eliminating the need for extensive feature engineering and achieving improved model performance through enhanced input initialization. Distinguishing itself from previous work, Sun et al. [31] employed the XGBoost algorithm [50] to pre-classify edge features, while Liu et al. [43] employed principal component analysis [51] to automatically capture feature representations and incorporated an attention mechanism after the LSTM to assign importance weights to different features. Additionally, Zhang et al. [52] leveraged heterogeneous networks to emphasize the significance of edge information. They employed heterogeneous networks to describe the representations of exercises and their attributes and added an attention mechanism after the RNNs model to determine the importance weights of different exercises.

Despite the advancements in knowledge tracing through multi-feature analysis, there remain areas that require further improvement. Firstly, the idea of incorporating these edge features to enhance the prediction accuracy of deep knowledge tracing in the educational domain lacks theoretical support. Secondly, there is a need to consider the inherent connections between different types of edge features. Lastly, most studies treat each feature as equally important, although some have attempted to address this issue by introducing attention mechanisms. However, these attention mechanisms are typically added after the RNNs models, primarily to handle long-term temporal dependencies, and their interpretability in terms of the importance of response features is limited. Nevertheless, multi-feature analysis in knowledge tracing has contributed to the field's development by characterizing learners' real learning processes and improving the predictive capabilities of deep knowledge tracing models.

3. Preliminaries

This section formalizes the research problem of MLFKT addressed in this paper. It introduces the definition of the LRR channel and presents two significant hypotheses within the education domain. For easy reference, Table 1 summarizes the symbols used throughout the paper.

Table 1. Summary of notations.

Notation	Description
S^i	Sequence of learning interactions for learner i up to moment t
X_t^i	Learning interaction of learner i at moment t
F_C	Learner–resource response channels
N	Number of LRR channels
f^l, L	Learner feature, number of learner features
f^e, E	Resource feature, number of resource features
f^r, R	Response feature, number of response features
q_t	Exercise performed by learners at moment t
r_t	Learners' response to a particular exercise at moment t
v_t	Feature vector of LRR channel at moment t
F_C^N	Feature vector containing N LRR channels up to moment t
$V_t^{'}$	Feature vector of LRR channel with attention mechanism weighting at moment t
v_t^{att}	Feature vector with attention weight
r_{t+1}	Predicted learners' response at moment $t+1$

3.1. Definitions

In this section, we provide five precise definitions of the proposed method, which builds upon two following educational a priori hypotheses:

Hypothesis 1 (Psychometric theory)**.** *The learner's mental psychological state can be inferred by observing their external behavior, and this mental state also influences their behavioral performance.*

Hypothesis 2 (Item response theory)**.** *Learners' responses and performance on a given resource are specifically associated with their underlying knowledge structure.*

When learners interact with an online learning platform, their interactions generate learning logs that contain a wealth of information about their responses. These logs primarily consist of three categories: learner features, resource features, and response features. Let us define the following:

Definition 1 (Learner features)**.** *Learner features are a collection of characteristics related to learner identity and learning ability. F_L represents the set of learner features, which encompasses L distinct learning features.*

Definition 2 (Resource features)**.** *Resource features are descriptions of the content and related attributes of the learning materials used by learners. F_E represents the set of resource features, including E different resource features.*

Definition 3 (Response features)**.** *Response features represent the interaction characteristics between learners and learning resources. F_R represents the set of response features, comprising R response features.*

$F_L \times F_E \times F_R = \left\{ (f^l, f^e, f^r) | (f^l \in F_L) \wedge (f^e \in F_E) \wedge (f^r \in F_R) \right\}$ denotes the Cartesian product of the sets of learner features, resource features, and response features.

Definition 4 (Mental knowledge structure)**.** *From the above two hypotheses, the external behavior features, which are learner features, resource features, and response features, are acquired to model the learner's knowledge states and thus reflect the learner's intrinsic knowledge structure.*

Based on these hypotheses, we posit that learner features, resource features, and response features exert influence on the learners' learning process, and there exists a distinct relationship among these features. Hence, we introduce the concept of the LRR channel.

Definition 5 (LRR channel)**.** *In accordance with the aforementioned educational hypotheses, each feature set $\left\{ f^l, f^e, f^r \right\}$ in $F_L \times F_E \times F_R$ that satisfies the hypotheses is defined as an LRR channel. The total number of such LRR channels is denoted by N, where $l \in (1, L)$, $e \in (1, E)$, $r \in (1, R)$, and $N \leq (L \times E \times R)$.*

3.2. Problem Definition

The objective of knowledge tracing is to model the evolution of learners' knowledge states and forecast their ability to correctly answer future exercises. In this work, we propose a novel deep knowledge tracing model that incorporates deep multi-feature fusion. Our model takes into account various factors that influence the learning process, including learner features, resource features, and response features. Moreover, it establishes connections between these features by constructing LRR channels based on educational priors.

Let $S^i = (X_1^i, X_2^i, \cdots, X_t^i)$ represent the sequence of learning interactions for learner i up to moment t. Here, i denotes the learner index, X_t denotes the learner's interaction at moment t, and X_t comprises N LRR channels. The learner's response to exercise q_t

at moment t is denoted as r_t, with $r_t = 1$ indicating a correct response and otherwise indicating an incorrect response.

Definition 6 (Learner performance prediction)**.** *We aim to predict the probability $P(r_{t+1} = 1|q_{t+1}, X_t)$ of the learner answering exercise q_{t+1} correctly at moment $t+1$ based on the learner's historical learning interactions S^i up to moment t.*

4. Methods

This paper presents a novel knowledge tracing model framework called MLFKT, which draws inspiration from previous studies such as BKT [6], DKT [9], DKVMN [30], SAKT [37], and MDKT [43]. The MLFKT framework aims to enhance the tracing of learners' knowledge states and to improve the prediction of learning performance by incorporating multiple learning features.

The MLFKT framework consists of four components: LRR channel construction, LRR channel feature representation learning, multi-deep fusion of LLR features using an attention mechanism, and long-term knowledge state tracing using Bi-LSTM. The first component utilizes learners' learning logs from online learning platforms to capture factors that influence learners' knowledge states and performance. Subsequently, the LRR channels are constructed. In the feature representation learning phase of LRR channels, low-dimensional embedding representations of these channels are obtained using stacked auto-encoders. The multi-deep fusion of LLR features based on the attention mechanism assigns different weight ratios to different LRR channels to refine the more crucial information. For long-term knowledge state tracing, Bi-LSTM is employed to model the learner's knowledge state based on their historical learning interactions, effectively addressing the challenge of long-term dependencies in temporal sequences. Finally, a prediction layer is designed to forecast the learner's performance in the next learning interaction. The detailed implementation of the proposed MLFKT framework is illustrated in Figure 1.

4.1. Construction of the LRR Channels

This subsection consists of two parts: (1) acquisition of learning features and (2) construction of LRR channels.

4.1.1. Acquisition of Learning Features

Psychometric theory suggests that learners' inherent potential can be inferred by analyzing their external responses. The objective of knowledge tracing is to track changes in learners' knowledge states based on their historical responses to resources and predict their subsequent responses. However, traditional knowledge tracing approaches typically define the external response solely as the learner's response to an exercise, which does not fully capture the complexity of real learning scenarios. For instance, learners with longer response times to a resource may indicate weaker mastery compared to those with shorter response times. Therefore, in this work, we aim to collect and analyze learners' learning logs from online learning platforms, which contain valuable information about learners' responses to various resources. These logs are divided into three primary categories: learner features, resource features, and response features, as shown in Table 2. Learner features encompass personal and cognitive information about the learners, while resource features include information about the resources and derived information from them. Response features capture the learners' behaviors resulting from their interactions with the resources. These features have a significant impact on tracing changes in learners' knowledge states and predicting their performance. Thus, in our subsequent work, we endeavor to incorporate multiple features, including learner features, resource features, and response features, to enhance the knowledge tracing process.

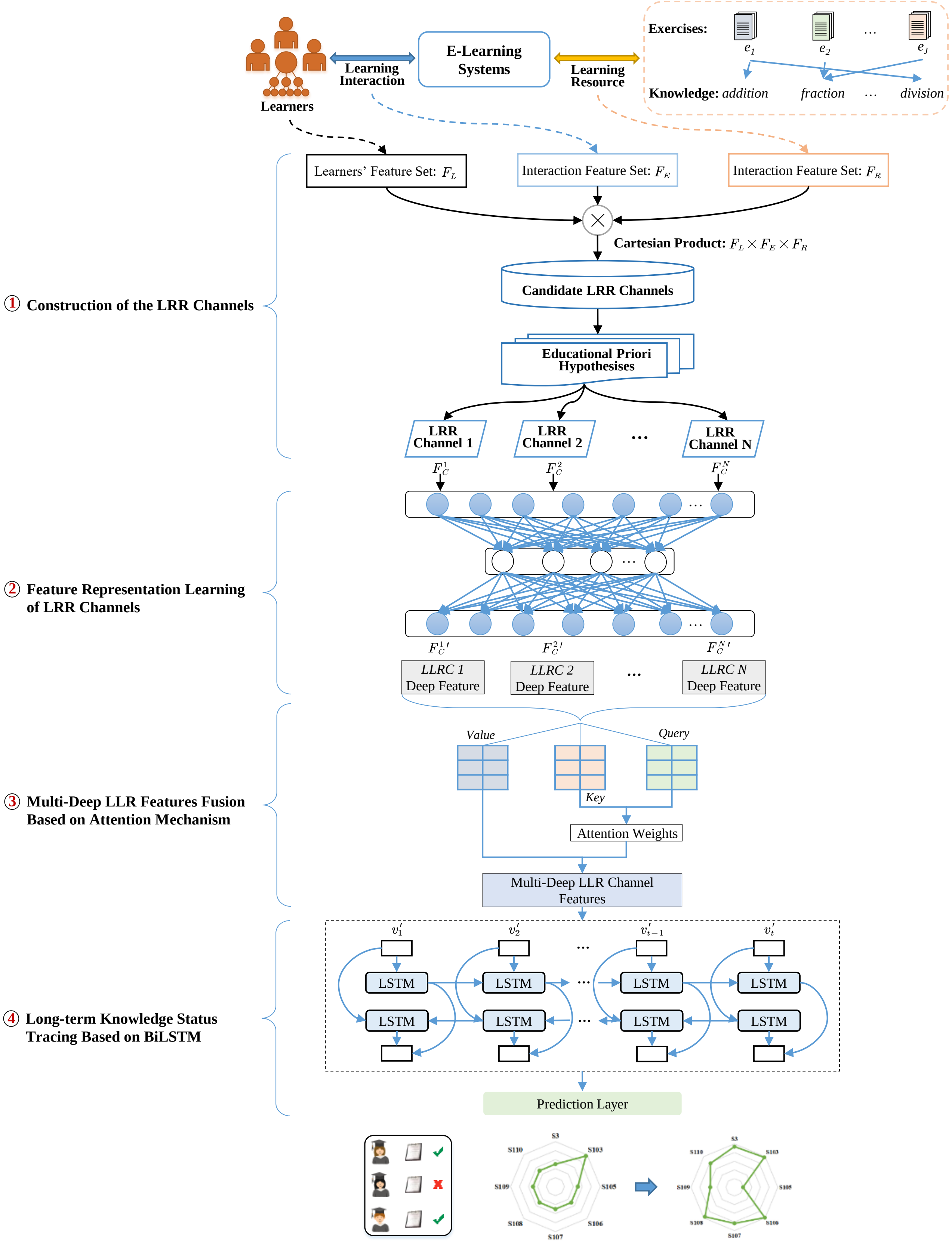

Learners
Learning Interaction
E-Learning Systems
Learning Resource
Exercises:
e_1
e_2
e_J
Knowledge: *addition* *fraction* … *division*
Learners' Feature Set: F_L
Interaction Feature Set: F_E
Interaction Feature Set: F_R
Cartesian Product: $F_L \times F_E \times F_R$
Candidate LRR Channels
Educational Priori Hypothesises
LRR Channel 1
LRR Channel 2
LRR Channel N
F_C^1
F_C^2
F_C^N
$F_C^{1\prime}$
$F_C^{2\prime}$
$F_C^{N\prime}$
LLRC 1 Deep Feature
LLRC 2 Deep Feature
LLRC N Deep Feature
Value
Query
Key
Attention Weights
Multi-Deep LLR Channel Features
v_1'
v_2'
v_{t-1}'
v_t'
LSTM
Prediction Layer
① Construction of the LRR Channels
② Feature Representation Learning of LRR Channels
③ Multi-Deep LLR Features Fusion Based on Attention Mechanism
④ Long-term Knowledge Status Tracing Based on BiLSTM
S3
S103
S105
S106
S107
S108
S109
S110

Figure 1. The proposed MLFKT framework encompasses four key components, namely (**1**) LRR channel construction, (**2**) LRR channel feature representation learning, (**3**) multi-deep fusion of LLR features using attention mechanism, and (**4**) long-term knowledge state tracing using Bi-LSTM.

Table 2. The summarization of learner features, resource features, and response features.

Type	Learning Feature	Related Information	Description
Learner Features	Grade [53]	Personal Information	Characteristics of the learner's grade level
	Ability [3]	Cognitive Information	Learners' potential subjects' abilities
Resource Features	Knowledge points [5]	Exercise Information	The knowledge contained in exercises
	Difficulty [26]	Exercise Information	The difficulty of exercises
Response Features	Correctness [21]	Behavior Information	Learners' responses to exercises
	Hints [47]	Behavior Information	Whether the learner sought official tips
	Response Time [43]	Behavior Information	Time spent by learners answering
	Attempts [54]	Behavior Information	Times of learner correctly answering

Deep knowledge tracing models that incorporate multiple learning features have gained attention for their ability to consider various factors that influence learners' knowledge states and performance. However, these models often treat these features as randomly matched or uniformly processed, leading to a lack of interpretability within the educational domain and overlooking the intrinsic connections between the features. For instance, KTMFF [26] utilizes a multi-headed attention mechanism to integrate learning features such as knowledge point difficulty, learner ability, and response duration. While the model is data-driven and effective, it fails to provide an explanation for the underlying connections between these different learning features.

Item response theory, on the other hand, suggests that subjects' responses and performance on test items are specifically related to their underlying features, highlighting the inherent link between features. In order to establish this intrinsic link, we have developed a LRR channel based on item response theory. However, we believe that the learners' potential knowledge states are not solely determined by their responses to a given resource. Instead, they are also influenced by additional tapping information about the resource, as well as the learners' personal and cognitive characteristics. Furthermore, the responses themselves should not be limited to simply correct or incorrect answers to exercises.

4.1.2. Construction of LRR Channels Based on Educational Hypotheses

To enhance the understandability and training effectiveness of the model, we perform feature engineering on the factors influencing learners' knowledge states and performance before constructing LRR channels. This process involves transforming these features into categorical features, which we refer to as class features. The main objective of class features is to discretize the input learner features, resource features, and interaction features into numerical intervals, simplifying the complexity of the model input.

For instance, consider response duration time, which is a continuous variable. By dividing it into intervals and categorizing it into six classes, we create class features for response duration time. This approach aligns with the actual learning scenarios. During data collection, when recording response times of learners answering exercises, we observe significant differences in response times, ranging from 10 to 100 s. However, a mere 1 s difference between a learner who took 10 s and another who took 9 s may not necessarily indicate varying levels of knowledge mastery. There might be an underlying similarity between them. Therefore, we consider response times within specific ranges as categories and transform them into class features. Similar treatment can be applied to the remaining features listed in Table 2 to transform them into class features. This approach effectively reduces the complexity of the input learning features, facilitating a more manageable representation of the educational data.

Guided by the two educational a priori hypotheses derived from psychometric theory and item response theory, we select feature sets from the Cartesian product of the transformed learner features, resource features, and response features. These feature sets represent individual LRR channels, aligned with the hypotheses.

By leveraging feature engineering and the categorization of features into class features, we simplify the representation of complex input features, ensuring a more interpretable and effective learning model. These steps contribute to the overall understandability and training efficiency of the model, aligning with educational theories and enhancing its applicability in real-world learning scenarios.

4.2. Feature Representation Learning of LRR Channels

As discussed in the previous subsection, the LRR channels proposed in this paper incorporate a diverse range of features that influence learners' knowledge states and performance. However, these LRR channels often contain redundant and irrelevant information at the feature level. Additionally, as the number of features increases, problems such as high-dimensional input feature vectors, longer training times, increased parameters, and higher computational costs arise. These challenges can make it difficult for the model to converge to the global optimum and can adversely affect prediction accuracy. Therefore, finding an effective approach to extract low-dimensional embeddings from LRR channels and increase their density is crucial.

The introduction of stacked auto-encoders (SAEs) provides an ideal solution to this problem. An auto-encoder is an unsupervised feature learning algorithm that trains an artificial neural network to reconstruct the input data itself. It generates a compact representation of the main features contained in the input data. A stacked auto-encoder consists of multiple individual auto-encoders, allowing for the learning of embedding feature vectors at different dimensions and levels from complex and high-dimensional input data. The training process of stacked auto-encoders consists of two steps: pre-training and fine-tuning. In the pre-training step, each individual auto-encoder is trained using unsupervised methods. The output of one layer becomes the input for the next layer, and this process continues until all the hidden layers are trained. The fine-tuning step involves training the entire stacked auto-encoder and optimizing the weights and biases using error back-propagation. In summary, each hidden layer in the stacked auto-encoder provides an alternative representation of the input feature values, capturing different dimensions and levels of embedding feature vectors from complex and high-dimensional input data without losing important information. Furthermore, it reduces the dimensionality of the input data, enabling deep feature extraction. Therefore, we utilize stacked auto-encoders to obtain low-dimensional embedding representations for each LRR channel.

Before being fed into the SAE, the LRR channels undergo cross-feature processing. In the 2010 KDD Cup competition [55], it was observed that cross-featured representations improved model prediction performance, while separate feature representations showed a decline in performance. The cross-feature processing method is defined as follows:

$$C(\left\{f_t^l, f_t^e, f_t^r\right\}) = f_t^l + (\max(f^l) + 1) \times (\max(f^e) + 1) \times f_t^r \tag{1}$$

where $\left\{f_t^l, f_t^e, f_t^r\right\}$ represents the interaction among learners at moment t within an LRR channel composed of learner feature l, resource feature e, and response feature r, which adheres to a specific educational domain hypothesis. For instance, if $f_t^l = a_t, f_t^e = q_t, f_t^r = r_t$, then f_t^l denotes the learner's ability at moment t, f_t^e represents the learner's practice response at moment t, f_t^r indicates whether the learner answered correctly at moment t, $\max(f^l)$ signifies the maximum value among the data for all ability features, $\max(f^e)$ denotes the maximum value among the data for all practice features, and $C(\cdot)$ symbolizes the crossover feature. Subsequently, the LRR channels undergo one-hot encoding to produce a fixed-length vector, which represents a feature vector of LRR channels following the educational domain rules encompassing learner feature l, resource feature e, and response feature r. This encoding process is achieved using the following formula:

$$v_t = O(C(\left\{ f_t^l, f_t^e, f_t^r \right\})) \tag{2}$$

Here, $O(\cdot)$ represents the one-hot encoding, and v_t represents the feature vector that contains the LRR channels formed by the learner's interaction with the exercise at moment t.

In the knowledge tracing task, the learner interacts with the exercise from moment 1 to t, resulting in the generation of feature vectors $V = (F_C^1, F_C^2, \ldots, F_C^N)$ that consist of N different LRR channels. Specifically, $F_C^N = (v_1, v_2, \ldots, v_t)^N$ represents the feature vector for the Nth LRR channel from moments 1 to t. These feature vectors, containing distinct LRR channels, are then individually fed into the SAEs to extract low-dimensional embedding representations of these LRR channels, thereby enhancing the density of each LRR channel. We will use F_C^N as an example to illustrate the functioning of the SAE.

The first auto-encoder of the SAEs takes F_C^N as input and produces the output $\tilde{y}^N$ using the following equations:

$$\begin{aligned} h_{N1} &= \tanh(W_{N1} F_C^N + b_{N1}) \\ \tilde{y}^N &= \tanh(W_{N1}^T h_{N1} + b_{N2}) \end{aligned} \tag{3}$$

Here, h_{N1} represents the hidden layer of the first encoder, W_{N1} is the weight matrix connecting the input layer and the hidden layer of the first encoder, and b_{N1} and b_{N2} are the corresponding bias vectors. The activation function tanh is used in these equations.

The feature $\tilde{y}^N$ obtained from the first auto-encoder serves as the input for the second auto-encoder of the SAE. The output $\tilde{z}^N$ is generated by the second auto-encoder using the following equations:

$$\begin{aligned} h_{N2} &= \tanh(W_{N2} \tilde{y}^N + b_{N3}) \\ \tilde{z}^N &= \tanh(W_{N2}^T h_{N2} + b_{N4}) \end{aligned} \tag{4}$$

In these equations, h_{N2} represents the hidden layer of the second encoder, W_{N2} is the weight matrix connecting the output layer of the first encoder and the hidden layer of the second encoder, and b_{N3} and b_{N4} are the corresponding bias vectors. The activation function tanh is used as well.

Since the SAE is trained jointly with the Bi-LSTM as mentioned later, fine-tuning the SAE weights increases the number of parameters and thus provides a better fit. Therefore, the SAE is pre-trained separately. The weights of the parameters are updated using the Adam method, and the objective function for optimization is the squared reconstruction error:

$$l = \frac{1}{2} \sum_{t=1}^{T} (\tilde{z}^N - v_t^N)^2 \tag{5}$$

Here, v_t^N represents the actual output, and $\tilde{z}^N$ represents the expected output. The output of the hidden layer of the second auto-encoder, obtained from the trained SAE, serves as the low-dimensional embedding representation vector of the LRR channel.

4.3. Multi-Deep LLR Features Fusion Based on Attention Mechanism

Upon reviewing existing studies on knowledge tracing using multi-feature analysis, we observed that most researchers either treat each feature equally or introduce an attention mechanism after modeling the learner's interaction sequence. However, the former overlooks the varying degrees of influence that different features may have on learner performance, while the latter lacks interpretability regarding feature importance. Additionally, the inclusion of an attention mechanism often primarily increases the temporal dependence between sequential elements, without directly extracting the importance of different features. In this paper, to enhance the specificity and effectiveness of LRR channels, we propose incorporating an attention mechanism to assign different weights to each LRR channel before inputting them into the RNNs model. This approach allows the model to concentrate on crucial information and to learn to absorb it fully, thereby facilitating the comprehensive integration and deep correlation of features.

As described in the preceding subsection, the low-dimensional embedding representation vector of LRR channels, obtained through SAE, serves as the input for the attention mechanism. Consequently, the input feature vector for the attention mechanism at time t is given by:

$$V_t' = W_2 V_t + b_2 = (v_t^{1'}, v_t^{2'}, \cdots, v_t^{N'}) \tag{6}$$

Here, V_t represents the feature vector containing N LRR channels generated at time t, and $V_t' = (v_t^{1'}, v_t^{2'}, \cdots, v_t^{N'})$ denotes the input feature vector for the attention mechanism, encompassing N LRR channels at time t. $W_2 = (W_{12}, W_{22}, \cdots, W_{N2})$ signifies the pre-trained weight of SAE, while $b_2 = (b_{12}, b_{22}, \cdots, b_{N2})$ represents the corresponding bias vector. Notably, since SAE is trained separately, the weights employed for constructing the attention mechanism feature vectors remain unchanged and rely on the pre-trained weights.

Following the feature input, the attention distribution α_t for the LRR channel is computed as:

$$\alpha_t = \sigma(U_{att} \times \tanh(W_{att}[v_t^{1'}, v_t^{2'}, \cdots, v_t^{N'}] + b_{att})) \tag{7}$$

Here, W_{att} corresponds to the weight of the attention module, while b_{att} represents the associated bias vector. σ and tanh denote the activation functions.

Upon obtaining the attention distribution for the LRR channel, the input feature vector of the attention mechanism is encoded to produce the feature vector v_t^{att} carrying the attention weights:

$$v_t^{att} = \alpha_t[v_t^{1'}, v_t^{2'}, \cdots, v_t^{N'}] + [v_t^{1'}, v_t^{2'}, \cdots, v_t^{N'}] \tag{8}$$

4.4. Long-Term Knowledge Status Tracing Based on Bi-LSTM

The conventional deep knowledge tracing method relies on RNN, which effectively handles temporal data and has been widely employed by researchers in knowledge tracing tasks. However, RNN encounters difficulties when dealing with information spanning long time intervals. During training, it is susceptible to issues such as gradient disappearance and gradient explosion. To overcome these limitations, researchers have turned to LSTM, a variant of RNN. LSTM addresses the problem of long-term dependency by incorporating three carefully designed gates: the forgetting gate, the input gate, and the output gate. The forgetting gate determines the importance of information from the previous hidden layer state, the input gate determines the importance of information from the current state, and the output gate determines the next hidden layer state. Consequently, researchers have increasingly utilized LSTM and other RNN variants such as GRU in knowledge tracing tasks to predict learners' future performance.

However, online learning platforms often accumulate learners' learning logs over extended periods, resulting in exceedingly long sequences of learning interactions that surpass the capabilities of LSTM to handle prolonged time sequences. Moreover, existing approaches only consider the influence of memory factors accumulated during past learning on learners' future performance, overlooking the dynamic nature of learners' knowledge state and performance, shaped by both memory and forgetting factors. According to the learning curve theory [56], continuous practice enables learners to eventually master the relevant knowledge. German psychologist Ebbinghaus discovered that forgetting begins immediately after learning and increases over time. Therefore, the occurrence of forgetting during the learning process significantly affects learners' knowledge states and performance.

This challenge can be addressed by leveraging Bi-LSTM [22], which is an extension of traditional LSTM. As shown in Figure 1, the Bi-LSTM introduced for modeling consists of two LSTMs superimposed on each other: one processes the input sequence in the forward direction, while the other processes it in the backward direction. The output is determined by combining the states of both LSTMs. At each time step, six distinct weights are employed, corresponding to the input to the forward and backward hidden layers $(W1, W3)$, the hidden layer to itself $(W2, W5)$, and the forward and backward hidden layers to the output layer $(W4, W6)$. Notably, there is no information flow between the forward and backward hidden layers, ensuring an acyclic unfolding graph. As shown in Figure 1, the forward layer is computed in the forward direction from time steps 1 to t, with the output of the forward hidden layer saved at each time step. Similarly, the backward layer calculates the output of the backward hidden layer by traversing the time steps from t to 1. Finally, the results of the forward and backward layers at corresponding time steps are combined to obtain the final output.

We select bidirectional LSTM to model learners' knowledge states and predict their future performance for two primary reasons:

1. Learners' learning processes are gradual and evolve slowly, necessitating a comprehensive tracking of their knowledge states that takes into account the influence of time series on prediction results. Bidirectional LSTM exhibits heightened sensitivity to long-term temporal information.
2. Learning processes are influenced by both memory and forgetting factors. While considering the impact of learners' memory accumulation on knowledge states, as reflected by past response information, it is crucial to also account for the influence of learners' forgetting degree on knowledge states, as indicated by forward response information. Bidirectional LSTM effectively leverages both past and forward response information, better characterizing learners' learning processes.

In the proposed approach, the feature vector v_t^{att} containing attention weights serves as the input to the Bi-LSTM model. The model is trained to effectively capture the learner's knowledge state and to predict their response r_{t+1} at moment $t+1$ through the following steps:

4.4.1. Processing of the First Layer

At moment t, the input to the first layer of the LSTM model determines the handling of information:

Step 1: Identify information to discard:

$$e_t = \sigma(E_e \cdot [h_{t-1}, v_t^{att}] + m_e) \tag{9}$$

Step 2: Determine values to update:

$$i_t = \sigma(E_i \cdot [h_{t-1}, v_t^{att}] + m_i) \tag{10}$$

Step 3: Decide which information to update:

$$\tilde{U}_t = \sigma(E_U \cdot [h_{t-1}, v_t^{att}] + m_U) \tag{11}$$

Step 4: Update the cell state:

$$U_t = e_t \otimes U_{t-1} + i_t \otimes \tilde{U}_t \tag{12}$$

Step 5: Output information related to the learner's knowledge acquisition:

$$\begin{aligned} g_t &= \sigma(E_g \cdot [h_{t-1}, v_t^{att}] + m_g) \\ h_t^1 &= g_t \cdot \tanh(U_t) \end{aligned} \tag{13}$$

In the above equations, E_e, E_i, E_U, E_g represent weight matrices, and m_e, m_i, m_U, m_g correspond to bias vectors. The activation functions σ and tanh are employed.

4.4.2. Processing of the Second Layer

Subsequently, the output from the first layer is passed to the second layer of the LSTM model to obtain the learner's knowledge mastery output:

$$\begin{aligned} e'_t &= \sigma(E_{e'} \cdot [h_{t-1}, v_t^{att}] + m_{e'}) \\ i'_t &= \sigma(E_{i'} \cdot [h_{t-1}, v_t^{att}] + m_{i'}) \\ \tilde{U}'_t &= \sigma(E_{U'} \cdot [h_{t-1}, v_t^{att}] + m_{U'}) \\ U'_t &= e'_t \otimes U'_{t+1} + i'_t \otimes \tilde{U}'_t \\ g'_t &= \sigma(E_{g'} \cdot [h_{t-1}, v_t^{att}] + m_{g'}) \\ h_t^2 &= g'_t \cdot \tanh(U'_t) \end{aligned} \tag{14}$$

Similarly, $E_{e'}, E_{i'}, E_{U'}, E_{g'}$ are weight matrices, and $m_{e'}, m_{i'}, m_{U'}, m_{g'}$ represent bias vectors. The activation functions σ and tanh are utilized.

The learner's final knowledge mastery is obtained by considering the knowledge mastery outputs from both layers of the LSTM model. Thus, the potential knowledge mastery formed by the learner's past learning trajectory at moment t can be expressed as:

$$h_t = [h_t^1, h_t^2] \tag{15}$$

Once the learner's knowledge mastery is acquired, the prediction of the learner's response r'_{t+1} is accomplished by integrating the learner's interaction with the provided resource at moment $t + 1$. This is achieved using the following equation:

$$r'_{t+1} = \sigma(E_{xy} X_{t+1} + E_{hy} h_t + m_{hy}) \tag{16}$$

Here, E_{xy} and E_{hy} denote weight matrices, and m_{hy} represents the corresponding bias vector. The activation function σ is applied to produce the final predicted response.

4.5. Optimization and Training

The objective of the model is to predict the performance of the next interaction, and to achieve this, the target prediction is extracted and the weights are updated using stochastic gradient descent. The loss function L is defined based on the predicted output r'_t and the target output r_t as follows:

$$L = -\sum_{t=1}^{T} (r'_t \log r_t + (1 - r'_t) \log (1 - r_t)) \tag{17}$$

To ensure smoother prediction results, l_1 and l_2 regularization terms are added to the loss function. The modified loss function L' is obtained by incorporating these regularization terms. The coefficients of the two regularization terms are denoted by λ_1 and λ_2, respectively. The regularization terms are defined as follows, and the optimization process of the objective function is illustrated in Algorithm 1.

$$\begin{aligned} L' &= L + \lambda_1 l_1 + \lambda_2 l_2 \\ l_1 &= \frac{\sum_{t=1}^{T-1} ||r'_{t+1} - r'_t||_1}{T-1} \\ l_2 &= \frac{\sum_{t=1}^{T-1} ||r'_{t+1} - r'_t||_2^2}{T-1} \end{aligned} \tag{18}$$

Algorithm 1: Optimization process of proposed MLFKT framework with the refined loss function

```
Input: Input: learning interaction sequence $S$
Initialization: weights $W_{l_1}, W_{l_2}$
for $e$ = 1: $E$ ($E$ = epoch) do
    Learning rate $\eta$ decreases to $1/10$ of the original $\eta$ per 30 epochs.
    for $t$ = 1: $T$ ($T$ = samples/batch size) do
        Calculate the overall loss:
            $\begin{cases} L' = L + \lambda_1 l_1 + \lambda_2 l_2 \\ L = -\sum_{t=1}^{T}(r'_t \log r_t + (1 - r'_t)\log(1 - r_t)) \\ l_1 = \frac{\sum_{t=1}^{T-1} ||r'_{t+1} - r'_t||_1}{T-1} \\ l_2 = \frac{\sum_{t=1}^{T-1} ||r'_{t+1} - r'_t||_2^2}{T-1} \end{cases}$
        Compute the backpropagation error: $\frac{\partial L(t)}{\partial W}$.
        Update the parameters:
            $\begin{cases} W_{l_1}(t+1) \leftarrow W_{l_1}(t) - \frac{\eta\lambda_1}{n} sgn(W_{l_1}(t)) - \eta\frac{\partial L(t)}{\partial W_{l_1}(t)} \\ W_{l_2}(t+1) \leftarrow W_{l_2}(t) - \eta(\frac{\partial L(t)}{\partial W_{l_2}(t)} + \frac{\lambda_2}{n} W_{l_2}(t)) \end{cases}$
    end
end
Return: $W_{l_1}, W_{l_2}$
```

5. Experiments

This section presents the evaluation of the learning performance prediction and interpretability achieved by our proposed MLFKT framework. Additionally, we discuss potential applications of our method in the field of smart education.

5.1. Experimental Datasets

Two datasets were utilized in this study: the publicly available ASSISTment2009 dataset (https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data (accessed on 26 November 2022)) and the SSAI dataset provided by the Squirrel AI Intelligent Adaptive Online Learning Platform (https://www.songshuai.com/ (accessed on 26 November 2022)). The ASSISTment2009 dataset originates from the "ASSISTments Skill Builder" problem set, which is a component of the ASSISTments intelligent learning system established in the United States in 2004. Due to its collection period from 2009 to 2010, it is commonly referred to as ASSISTment2009–2010. On the other hand, the SSAI dataset is obtained from the Squirrel AI Intelligent Adaptive Online Learning Platform, focusing on learners' interactions with middle school mathematics exercises. The fundamental characteristics of the ASSISTment2009 dataset and the SSAI dataset are summarized in Table 3. The other input learning features can be grouped into learner features, resource features, and response features, thus allowing for similar feature processing methods.

Table 3. Statistical summaries of ASSISTment2009 dataset and SSAI dataset. A check mark indicates that this information is included.

Basic Information	ASSISTment2009 Dataset	SSAI Dataset
Number of learner–exercise interactions	112,115	37,416
Number of knowledge	31	110

Table 3. *Cont.*

Basic Information		ASSISTment2009 Dataset	SSAI Dataset
Number of students		2626	9345
Learner Features	Grade		✓
	Ability		✓
Resource Features	Knowledge points	✓	✓
	Difficulty		✓
Response Features	Correctness	✓	✓
	Hints	✓	✓
	Response Time	✓	✓
	Attempts	✓	

Furthermore, this paper employed several preprocessing techniques on the two datasets to facilitate their utilization in the model. Firstly, any missing data pertaining to learners' behavior in both datasets were replaced with zeros to enable subsequent data coding. Secondly, interactions between learners and exercises that occurred three times or less were excluded from the dataset since the analysis focuses on temporal order. Additionally, any inconsistent or illogical data points, such as correctness feature values deviating from 0 or 1, were removed. Finally, duplicate data points present in both datasets were eliminated.

In the case of the ASSISTment2009 dataset, it encompasses various features including knowledge points, correctness, attempts, hints, response time, and ability. Among these, the number of attempts feature and the response time feature have wide value ranges, potentially introducing complexity to the model. Consequently, these features were transformed into class features. The highest recorded number of attempts in the ASSISTment2009 dataset was 3825, leading to the division of the attempts feature into three categories: 0 for no attempts, 1 for a single attempt, and 2 for more than two attempts. Initially measured in milliseconds, the response time feature in the ASSISTment2009 dataset was converted to seconds. Data points exceeding the threshold of 400 s were excluded from the dataset. Subsequently, the data underwent Z-Score distribution to attain a standardized normal distribution. Finally, the data were converted into class features. Values below -0.8 were classified as 0, values between -0.8 and 0.6 as 1, values between -0.6 and 0 as 2, and values greater than 0 as 3. The hints feature in the ASSISTment2009 dataset indicates the number of times learners accessed hints and, since the data values range exclusively from 0 to 10, this feature was more appropriately treated as a class feature. The ability feature in the ASSISTment2009 dataset was derived from an analysis employing the three-parameter IRT model. Originally spanning from -4 to 4, the feature was divided into seven categories, each representing a unit increment, resulting in a categorical representation.

Regarding the SSAI dataset, it encompasses knowledge points, correctness, difficulty, hint, response time, and ability features. The processing of response time features followed the same steps as applied to the ASSISTment2009 dataset. The view prompt feature in the SSAI dataset indicates whether learners viewed a specific prompt or not, resulting in two possible values: 1 for viewed and 0 for unviewed. The difficulty feature values range from 0 to 9, making it suitable for categorical representation. Similar to the ASSISTment2009 dataset, the ability feature in the SSAI dataset was also analyzed utilizing the three-parameter IRT model. Additionally, the knowledge points feature in the SSAI dataset was originally recorded in a character format but was converted to an integer type for improved data processing.

5.2. Baseline Models for Comparison

The evaluation encompasses three aspects: the effectiveness of learner–resource response channels, the effectiveness of a stacked auto-encoder and attention mechanism for representation learning, and the effectiveness of bidirectional LSTM for long-term modeling. Additionally, the proposed models are compared with six state-of-the-art methods: Bayesian knowledge tracing (BKT) [6], deep knowledge tracing (DKT) [9], DKT+ [29], dynamic key-value memory network (DKVMN) [30], self-attentive knowledge tracing (SAKT) [37], and multiple feature deep knowledge tracing (MDKT) [43].

The following baseline models are considered:

- **BKT [6]**: The BKT model aims to capture students' evolving knowledge state during knowledge acquisition. It involves monitoring learners' knowledge states over time and predicting their performance in subsequent learning interactions.
- **DKT [9]**: The DKT model explores the use of recurrent neural networks (RNNs) to model student learning. It offers the advantage of not requiring explicit encoding of human domain knowledge and enables the representation of more complex student knowledge.
- **DKT+ [29]**: The DKT+ model enhances the original DKT model by introducing regularization terms corresponding to reconstruction and waviness in the loss function. This regularization improves the consistency in prediction.
- **DKVMN [30]**: The DKVMN model leverages the relationships between underlying concepts and directly outputs a student's mastery level for each concept. It utilizes a static key matrix to store knowledge concepts and a dynamic value matrix to store and update the mastery levels of the corresponding concepts.
- **SAKT [37]**: The SAKT model aims to identify the relevant knowledge points from a student's past activities based on the given knowledge points and predicts mastery accordingly. It effectively handles data sparsity issues, as predictions rely on a limited set of past activities rather than relying solely on RNN-based methods.
- **MDKT [43]**: The MDKT model fully utilizes both student behavior features and exercise features. It combines the attention mechanism with the knowledge tracing model. Initially, machine learning is employed to automatically capture feature representations. Then, a fusion attention mechanism based on the recurrent neural network architecture is used for predicting student performance.

5.3. Evaluation Metrics

Knowledge tracing involves predicting learners' performance and encompasses both classification and regression aspects. In this paper, we utilize two evaluation metrics to assess the prediction performance of the models: area under the curve (AUC) and root mean square error (RMSE).

AUC is a widely adopted and reliable metric in classification problems. In this study, AUC represents the accuracy of the model's predictions, with correct learner responses considered as positive samples and incorrect responses as negative samples. The AUC value ranges from 0 to 1, where 0.5 indicates random prediction, while higher values indicate more accurate predictions.

For the regression aspect of the prediction task, we employ RMSE,

a commonly used metric for regression problems. RMSE provides a straightforward and interpretable measure of prediction performance without requiring adjustments based on specific application scenarios. In our experiments, a smaller RMSE value indicates better prediction performance, with values closer to 0 reflecting higher accuracy in predictions. The calculation of RMSE is as follows:

$$\mathrm{RMSE}(y_i, \hat{y}_i) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{19}$$

Here, m represents the size of the test dataset, y_i is the true value of the student's answer, and $\hat{y}_i$ is the predicted value of the student's answer.

5.4. Implementation Details

The experimental setup involves several development platforms and tools, including Tensorflow, a deep learning framework, Scikit-learn, a machine learning algorithm library, and Anaconda, a Python distribution with integrated scientific computing packages. Python was utilized for experiment design and implementation. The hardware specifications and scientific computing packages used are presented in Table 4. The main development environments employed in this experiment consisted of Windows 10, AMD Ryzen Threadripper 2950X 16-Core Processor, 32 GB RAM, GTX 1080Ti graphics card, Tensorflow 1.9, and Python 3.5.

Table 4. Software and hardware setup of the experiments.

Configuration Environment	Configuration Parameter
Operating System	Windows
Memory	8 GB
Programming Language	Python 3.5
Deep Learning Framework	Tensorflow 1.9
Python Libraries	Numpy, Pandas, Matplotlib

To identify the most effective hyperparameters, we conducted numerous experiments and empirically determined their optimal values. The optimal hyperparameters of these experiments can be found in Tables 5 and 6. The selection of these specific platforms, tools, and hardware configurations was guided by their compatibility, performance, and suitability for deep learning tasks. By utilizing these well-established and widely-used resources, we ensured a robust and reliable experimental setup, enhancing the credibility and reproducibility of our findings.

Table 5. Setting of SAE experimental parameters.

Batch Size	300
Learning Rate	0.005
Epochs	5

Table 6. Setting of Bi-LSTM experimental parameters.

Batch Size	300
Learning Rate	0.05
Epochs	5 × 40
Hidden Size	200

To ensure avoidance of overfitting when training the SAEs simultaneously with Bi-LSTM, pre-training of the SAE and fine-tuning of its weights are performed. Table 5 presents some of the experimental parameters used for training the SAE. The feature descent dimension is determined based on the original dimension of the features, as elaborated in the subsequent section. The weights of the parameters are trained using the Adam optimizer method, with the squared reconstruction error as the optimization objective. After training the SAE, the output of the second hidden layer, an auto-encoder of the SAE, serves as input to the attention mechanism, and the weights remain constant at the output due to pre-training. Subsequently, the output, a feature vector with weights, is fed into the Bi-LSTM model. Table 6 provides some of the experimental parameters for training the Bi-LSTM. The model is trained using the five-fold cross-validation method, employing a dropout rate of 0.5 and a learning rate of 0.05 to mitigate overfitting. The bifurcation entropy is utilized as the training target. Through experimentation, it was discovered that the Adam optimizer is less effective than the stochastic gradient descent method within the context of this paper. Therefore, the parameters are trained using the random gradient descent method.

6. Results

In this section, we present the experiment results and discuss the important findings from our experiments.

Comparison with State-of-the-Art Methods (RQ1)

To validate the effectiveness of the proposed framework in this paper, we conducted experiments using the ASSISTment2009 dataset and SSAI dataset. The training and test sets were divided into an 80% and 20% ratio, respectively, as described in the previous subsection. The specific experimental equipment and settings were also detailed earlier. All experiments were performed with five-fold cross-validation, and the final results were obtained by averaging the outcomes. The evaluation of experimental results was based on AUC and RMSE metrics, as explained in the previous subsection.

Previous studies on knowledge tracing have demonstrated that the DKT model with LSTM as the core model outperforms the DKT model with RNN and its variant GRU in terms of prediction accuracy.

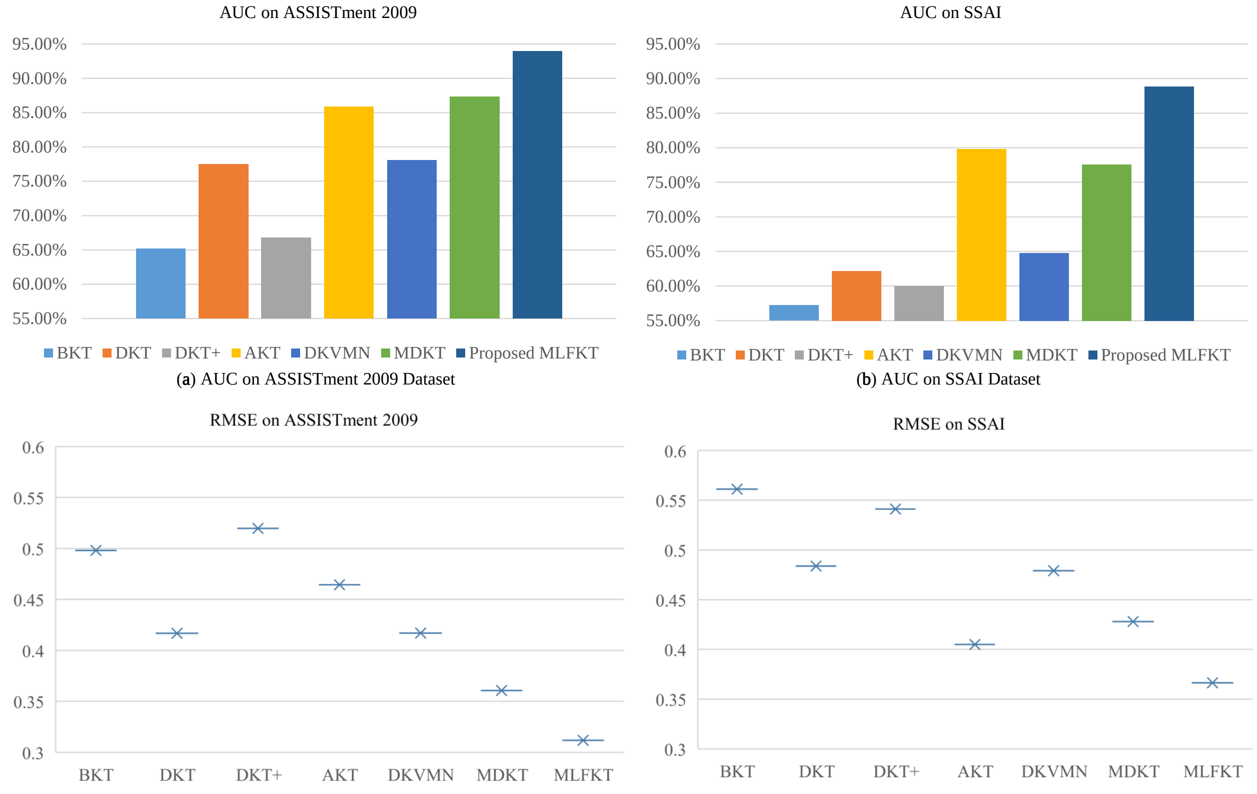
Therefore, in this paper, we adopted the DKT model with LSTM as the benchmark model for comparison with the proposed framework. The comparative results are presented in Table 7 and Figure 2.

As shown in Table 7, our proposed framework outperformed other classical knowledge tracing models in both evaluation metrics on the ASSISTment2009 dataset and SSAI dataset. This superiority highlights that our framework excels in predicting learner performance, benefiting from the consideration of multiple learning factors. Additionally, by constructing LRR channels to establish connections between features, employing SAE and attention mechanisms to represent these channels, and employing Bi-LSTM to model long time-series interaction sequences, we better characterize the learning process.

As shown in Figure 3, after five epochs of model training, the training and testing AUC values of the proposed method and MDKT exhibit similar performance, indicating that both algorithms converge and achieve relatively good results. However, there is a notable disparity between the training AUC of DKT and its testing AUC. Additionally, the proposed method converges with DKT after 15 epochs of training, while MDKT converges after 10 epochs. These findings suggest that the training complexity of the proposed method is comparable to DKT and slightly weaker than MDKT, but it remains within an acceptable range.

Table 7. The prediction performance of the proposed method is compared with six state-of-the-art methods on two real datasets. Bolded values indicate the best performers, and underlined ones indicate the previous best performers.

Model	ASSISTment2009		SSAI	
	AUC	RMSE	AUC	RMSE
BKT	65.18%	0.4982	57.23%	0.5611
DKT	77.45%	0.4169	62.15%	0.4838
DKT+	66.78%	0.5198	60.02%	0.5411
DKVMN	78.06%	0.4170	64.79%	0.4791
SAKT	85.86%	0.4644	<u>79.80%</u>	<u>0.4050</u>
MDKT	<u>87.36%</u>	<u>0.3606</u>	77.57%	0.4279
Proposed MLFKT	**93.92%**	**0.3118**	**88.81%**	**0.3659**



(**a**) AUC on ASSISTment 2009 Dataset

(**b**) AUC on SSAI Dataset

(**c**) RMSE on ASSISTment 2009 Dataset

(**d**) RMSE on SSAI Dataset

Figure 2. The proposed MLFKT framework was compared with six state-of-the-art methods. In this paper, six state-of-the-art KT methods were replicated, and experiments were conducted on two real datasets and were evaluated using two metrics, AUC and RMSE. A larger AUC indicates higher prediction accuracy, and a smaller RMSE indicates better stability of the methods.

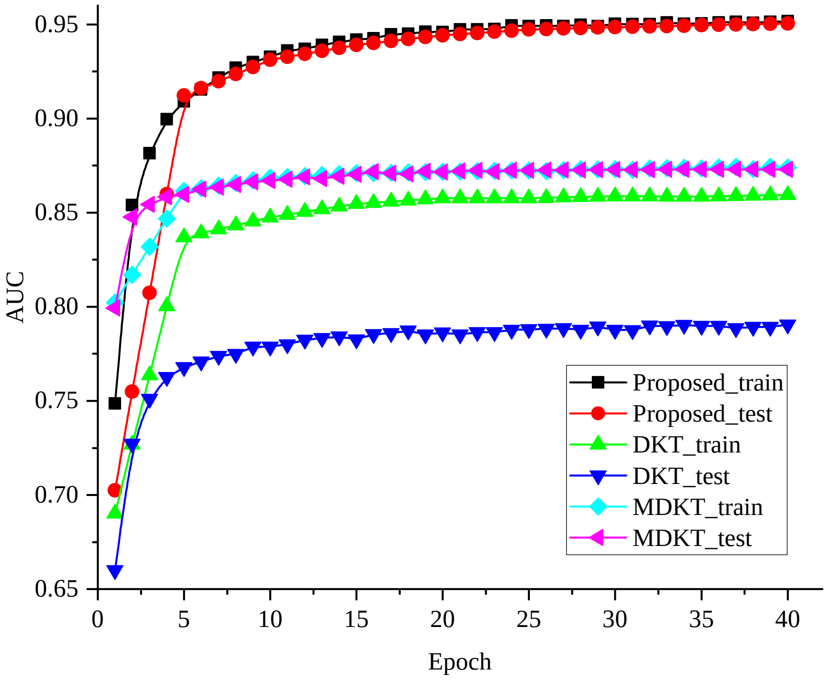


Figure 3. The training and testing AUC on the ASSISTment2009 dataset. Forty epochs of training and testing were implemented to verify the convergence characteristics and computational complexity of the proposed method, DKT, and MDKT.

7. Discussion

In addition to the aforementioned experiments, we conducted a series of ablation experiments to assess the effectiveness of the LRR channel, stacked auto-encoders, attention mechanisms, and bidirectional LSTM in our proposed framework. The experimental equipment, settings, and evaluation metrics remained consistent with the previous experiments. Each experiment was repeated 10 times, and the average results were reported. Since there are various ways to combine features, this paper focuses on discussing only a few representative cross-feature combinations, which are selected based on the principles of item response theory.

The experimental results are presented in Table 8. The "DKT-LSTM" model refers to the DKT model with LSTM as the core, while "MLFKT-X" represents the deep knowledge tracing model with Bi-LSTM as the core and incorporates different LRR channels. In the table, "SC" refers to the knowledge–correctness feature combination, "NC" refers to the ability–correctness feature combination, "NSC" represents the ability–knowledge–correctness channel (based on the hypothesis that learners' correctness and scores on exercises are related to their abilities), "NST" represents the ability–knowledge–response time channel (based on the hypothesis that learners' answer time and scores on exercises are related to their abilities), and "ALL" denotes the inclusion of all features in the dataset. Additionally, "CNN" indicates the use of convolutional neural networks, "SAE" indicates the use of stacked auto-encoders, and "ATT" indicates the use of attention mechanisms.

As shown in Table 8, the proposed framework outperformed the DKT-LSTM model in terms of AUC and RMSE metrics on both the ASSISTment2009 and SSAI datasets. By incorporating various cross-feature combinations, such as MLFKT-SC, MLFKT-SC + NC, MLFKT-NSC, MLFKT-NSC + NST, and MLFKT-ALL, the performance of the framework was further improved. The utilization of additional techniques, such as CNN and SAE, contributed to even better results. Ultimately, by incorporating SAE and attention mechanisms (SAE + ATT) into the MLFKT-ALL model, we achieved the best performance, surpassing all other models in terms of both AUC and RMSE metrics.

Table 8. Nine ablation experiments were conducted on the ASSISTment2009 and SSAI datasets. Bolded values indicate the best performers.

Model	ASSISTment2009		SSAI	
	AUC	RMSE	AUC	RMSE
DKT-LSTM	85.60%	0.3770	76.77%	0.4312
MLFKT-SC	88.31%	0.3567	81.22%	0.4185
MLFKT-SC + NC	91.57%	0.3287	83.17%	0.4095
MLFKT-NSC	91.67%	0.3296	85.13%	0.3961
MLFKT-NSC + NST	91.45%	0.3293	85.65%	0.3934
MLFKT-ALL	90.76%	0.3379	85.90%	0.3991
CNN: MLFKT-ALL	86.80%	0.3685	83.86%	0.4039
SAE: MLFKT-ALL	93.84%	0.3125	88.39%	0.3690
SAE + ATT: MLFKT-ALL (Proposed)	**93.92%**	**0.3118**	**88.81%**	**0.3659**

7.1. Effectiveness of Multiple Features and LRR Channels (RQ2)

Incorporating various additional features that influence learners' knowledge states and performances and establishing intrinsic connections between these features through the construction of LRR channels can provide a better characterization of learners' learning processes. In our experiments on the ASSISTment2009 and SSAI datasets, we observed that incorporating more learner and resource features generally led to an increase in the AUC values and a decrease in the RMSE values. For example, on the SSAI dataset, the AUC value of MLFKT-NSC increased by 1.95% compared to MLFKT-SC, while the RMSE value decreased by 0.9%. Similarly, MLFKT-NSC + NST had an AUC increase of 0.52% and an RMSE decrease of 0.27% compared to MLFKT-NSC. Furthermore, MLFKT-ALL showed a 0.25% higher AUC value compared to MLFKT-NSC + NST. These results indicate that considering additional features that influence learners' knowledge states and performance can enhance the accuracy of performance prediction and provide better insights into learners' learning processes.

Moreover, we observed that MLFKT-NSC outperformed MLFKT-SC + NC in terms of both AUC and RMSE metrics, suggesting that constructing LRR channels can effectively enhance the associations between features. However, on ASSISTment2009, there were instances where incorporating more features did not lead to performance improvement and, in some cases, resulted in performance degradation. For example, the performance metrics of MLFKT-ALL on AUC and RMSE were better than those of MLFKT-SC but inferior to MLFKT-NSC. This discrepancy may be attributed to certain features negatively affecting learners' knowledge states and performance. Nevertheless, in general, incorporating additional features that influence learners' knowledge states and performance, as well as constructing LRR channels to enhance feature connections, can facilitate a more accurate modeling of learners' learning processes.

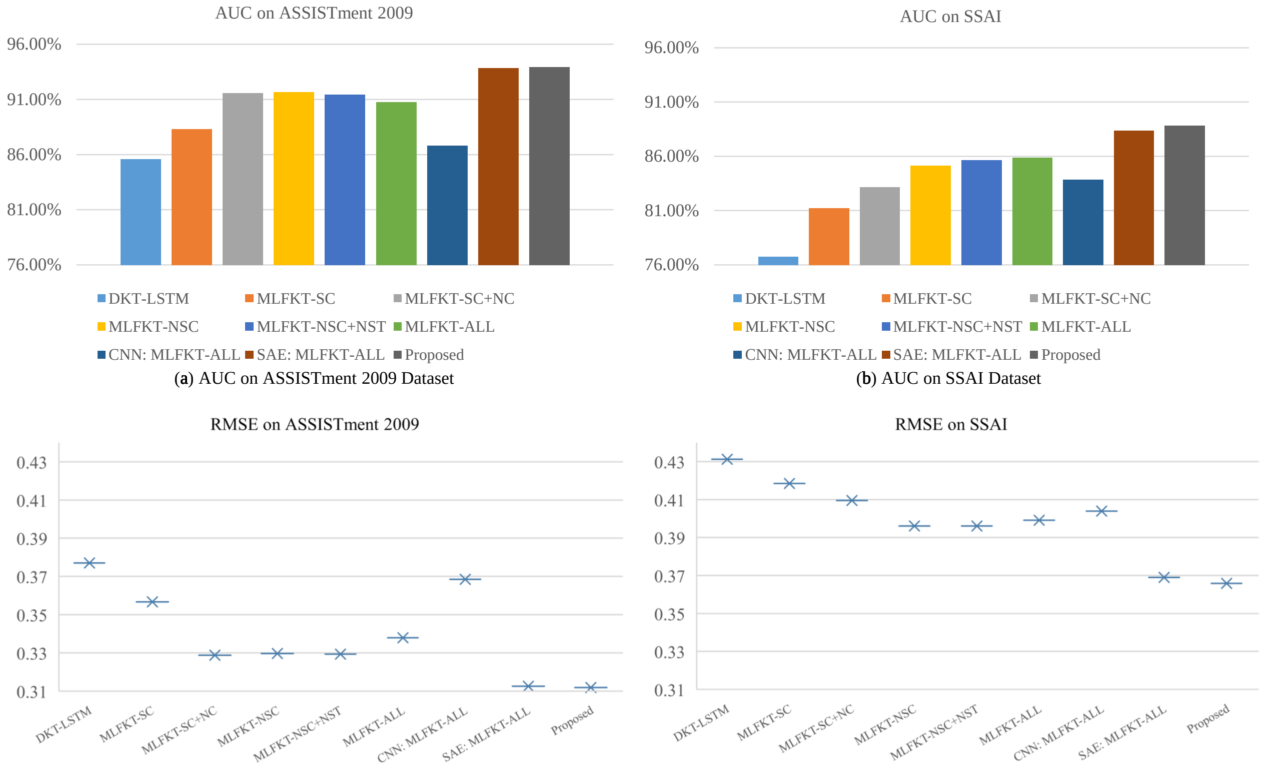
7.2. Effectiveness of Stacked Auto-Encoder and Attention Mechanism (RQ2)

The utilization of SAEs enables the extraction of low-dimensional embedding representations from different LRR channels, resulting in denser and more compact representations. By employing an attention mechanism to assign importance weights to various LRR channels, the model can focus on crucial information while fully integrating and characterizing features. Our experiments on the ASSISTment2009 and SSAI datasets demonstrate significant improvements in the accuracy of performance prediction when using SAE for representing learning of LRR channels. In ASSISTment2009, SAE: MLFKT-ALL achieved a 3.08% increase in AUC value compared to MLFKT-ALL, as shown in Figure 4. Notably, the representation learning achieved by SAE surpassed the effectiveness of using convolutional neural networks (CNN) on both datasets. Similarly, SAE + ATT: MLFKT-ALL outperformed SAE: MLFKT-ALL in terms of AUC and RMSE metrics on both datasets. The attention mechanism enables the model to focus on more important features during training, leading to improved feature representation. The results of our experiments demonstrate that the attentional weight coefficient for the NSC feature combination is higher than that of the NC feature combination. Conversely, the attentional weight coefficient for the NC feature combination is higher than that of the SC feature combination. This finding aligns with the results presented in Table 8, providing further support for our findings. In summary, the incorporation of SAE and the attention mechanism for representation learning enhances the integration and characterization of features, resulting in improved accuracy and effectiveness in predicting learner performance.

7.3. Effectiveness of Stacked Auto-Encoder and Attention Mechanism (RQ2)

The utilization of SAEs enables the extraction of low-dimensional embedding representations from different LRR channels, resulting in denser and more compact representations. Additionally, incorporating an attention mechanism assigns effectiveness weights to different LRR channels, allowing for a focus on more important information while facilitating the full fusion and characterization of features. Our experiments on the ASSISTment2009 and SSAI datasets reveal significant improvements in the accuracy of predicting learner performance through the use of SAE for representation learning of LRR channels. Specifically, SAE: MLFKT-ALL achieved a 3.08% improvement in AUC value compared to MLFKT-ALL in ASSISTment2009, as shown in Figure 4. Notably, the representation learning achieved by SAE surpassed the effectiveness of using convolutional neural networks (CNN), a finding consistent in the SSAI dataset. This improvement can be attributed to SAE's ability to extract embedding features of varying dimensions and levels from high-dimensional data, enabling deep feature extraction from input channels. Furthermore, SAE + ATT:

MLFKT-ALL outperformed SAE: MLFKT-ALL in terms of both AUC and RMSE metrics on both the ASSISTment2009 and SSAI datasets. The addition of the attention mechanism allows the model to focus more on crucial features during training, resulting in superior feature representation. In summary, employing SAE and the attention mechanism for representation learning of input channels enables a more comprehensive and profound integration and characterization of these features, thereby enhancing the accuracy and effectiveness of predicting learner performance.



(**a**) AUC on ASSISTment 2009 Dataset

(**b**) AUC on SSAI Dataset

(**c**) RMSE on ASSISTment 2009 Dataset

(**d**) RMSE on SSAI Dataset

Figure 4. In the experiments, nine ablation experiments were conducted on two real datasets and were evaluated using two metrics, AUC and RMSE. A larger AUC indicates higher prediction accuracy, and a smaller RMSE indicates better stability of the methods.

7.4. Effectiveness of Bidirectional LSTM (RQ2)

The classical education curve theory posits that learners experience a continuous cycle of memory formation and forgetting during the learning process. Therefore, incorporating memory and forgetting factors into the knowledge tracing model can improve the prediction of learner performance. By considering interaction sequences of learners over longer intervals and accounting for the influence of memory and forgetting, the Bi-LSTM enhances prediction accuracy. In ASSISTment2009, the AUC of MLFKT-SC improved by 2.71% and the RMSE decreased by 2.03% compared to the baseline. Similarly, in the SSAI dataset, MLFKT-SC exhibited a 4.45% improvement in AUC and a 1.27% decrease in RMSE compared to the baseline, as shown in Figure 4. This improvement stems from the ability of Bi-LSTM to effectively leverage the impact of accumulated learner memory, as reflected in past information, and to capture the influence of learner forgetting, as reflected in forward response information. Moreover, Bi-LSTM can effectively handle long sequences of learner interaction over time. Consequently, incorporating Bi-LSTM into the knowledge tracing task, considering both memory and forgetting factors, and modeling learners' long time-series interaction sequences, results in superior performance prediction.

The key contributions of this paper can be summarized as follows:

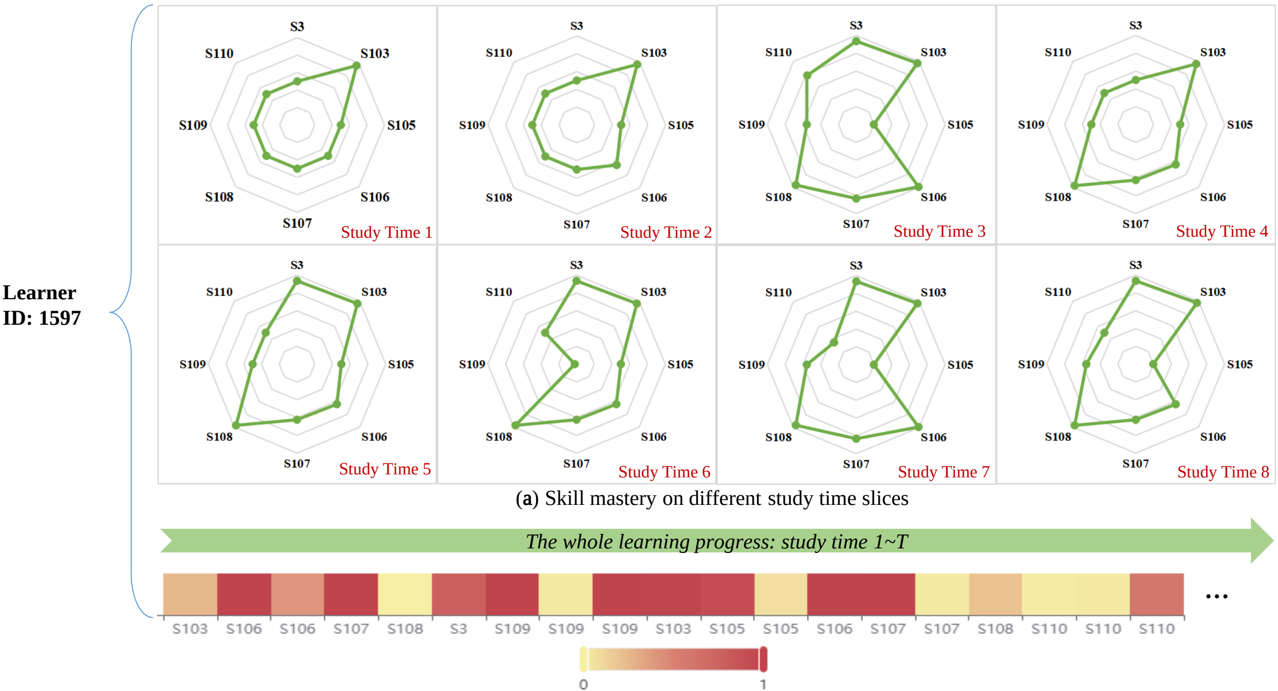
1. We propose a novel deep knowledge tracing model that incorporates learner features, resource features, and response features. LRR channels are constructed based on item response theory, following the principles of psychometric theory. The LRR channels enhance the intrinsic associations among different learning features and provide educational interpretability to the model.
2. Our proposed knowledge tracing framework based on LRR channels applies to a wide range of educational scenarios and can handle a large amount of learning information. We introduce a deep representation learning approach that leverages stacked autoencoders and attention mechanisms to fuse multiple features into the knowledge tracing task. We also introduce Bi-LSTM into the knowledge tracing task, which takes into account both memory and forgetting factors. By considering both past and future response information, Bi-LSTM comprehensively reflects the learner's memory accumulation and forgetting degree, thus enhancing the accuracy of knowledge state estimation.
3. The proposed framework outperforms six baseline methods on two real benchmark datasets, achieving state-of-the-art performance. Specifically, the proposed model improves the AUC by 12% over the previous best result on the knowledge tracing benchmark. This demonstrates the effectiveness of our proposed approach and its potential for practical applications in educational settings.

7.5. Applications in Education (RQ3)

Our framework offers valuable support to teachers in facilitating targeted instructional interventions and promoting focused learning among students. As depicted in Figure 5, the figure illustrates the predicted probability of a learner's correct response to an exercise alongside the corresponding changes in their cognitive knowledge structure.

By accurately predicting the likelihood of a learner providing a correct response, we can assess their mastery of specific knowledge points. This information proves instrumental in helping learners identify and address knowledge gaps, while empowering teachers to deliver more tailored interventions. By recommending appropriate answer sequences and learning resources to learners, teachers can optimize the learning process. The results of our aforementioned experiments not only showcase the framework's enhanced performance in predicting learner outcomes but also highlight its practical applicability within the education domain, addressing real-world challenges.

This research aims to investigate the mental dynamic cognitive states of learners' knowledge structure by modeling and analyzing their external behaviors, which include learner features, resource features, and response features. These mental states provide insights into the learners' internal knowledge structure and their cognitive processing, which are crucial for cognitive diagnosis. The novelty of this research lies in the expansion of the scope of application of the item response theory (IRT). Unlike traditional IRT, which only considers learners' correctness responses and utilizes a single ability parameter to describe their mental states, our approach not only incorporates a wider range of external learning features but also delves into the internal knowledge structure with greater granularity.



(**a**) Skill mastery on different study time slices

(**b**) Dynamic skill mastery sequences over the entire learning process

Figure 5. The knowledge features, such as S3, S103, S105, S106, etc., are defined in Table 2. This figure is primarily used to depict the learners' proficiency levels in various knowledge points and the dynamic progression of knowledge mastery as learning advances. (**a**) A radar chart is used to depict the learner's mastery of the eight knowledge points at a specific study time, with larger areas indicating higher levels of mastery. (**b**) The overall dynamics of the learner's knowledge mastery throughout the entire learning process, with darker colors representing greater proficiency. The radar chart in (**a**) is constructed by capturing a time-specific knowledge mastery from (**b**).

8. Conclusions

In this paper, we have proposed a novel knowledge tracing framework MLFKT, which is designed to enhance learning feature representation and improve learning performance prediction. Our framework offers three key contributions. Firstly, it leverages psychometric theory, which posits that learners' intrinsic features can be inferred from their extrinsic responses, allowing us to effectively utilize the abundant learning features available in online learning platforms. Additionally, we have constructed a learner–resource response channel, establishing intrinsic connections between features and expanding the applicability of item response theory to a broader range of educational scenarios, thereby enriching the processable information. Secondly, our framework incorporates SAEs and an attention mechanism to learn the representation of these channels. This approach enables the channels to acquire dense, low-dimensional embedding representations while retaining crucial information. Furthermore, by assigning varying effectiveness weights to different channels during fusion, we establish deep relationships and integration among the channels. Lastly, our framework is built upon the concept of LRR, taking into account the dual factors of memory and forgetting in learners' learning processes. By introducing Bi-LSTM into the knowledge tracing task, our model effectively captures the long-range interaction information among learners while better uncovering and portraying the entire learning trajectory. Experimental results demonstrate that our proposed knowledge tracing model surpasses previous models in accurately predicting learners' performances. In future work, we propose two avenues for further exploration. Firstly, we aim to develop more sophisticated models that can effectively capture the intrinsic knowledge structure of learners, thereby enhancing the modeling of their learning behaviors. Secondly, we plan to explore the integration of knowledge graphs as a means of providing better explanations from the perspective of structured knowledge modeling. Through the progress of this study, learners can develop a more comprehensive understanding of their knowledge structure and engage in more focused learning within online self-directed learning environments. This, in turn, enables teachers to provide more effective interventions and personalize the teaching and learning experiences, ultimately facilitating personalized education.

Author Contributions: Conceptualization, Z.W. and C.Z.; methodology, Z.W.; software, C.Z.; validation, Z.W., Y.H., S.Z. and R.Y.; formal analysis, Z.W. and C.Z.; investigation, Z.W., Y.H., S.Z. and R.Y.; resources, Z.W. and C.Z.; data curation, Z.W. and C.Z.; writing—original draft preparation, Z.W. and C.Z.; writing—review and editing, Z.W. and C.Z.; visualization, Z.W. and C.Z.; supervision, Z.W.; project administration, Z.W.; funding acquisition, Z.W. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The research work in this paper was supported by the National Natural Science Foundation of China (no. 62177022, 61901165, 61501199), the AI and Faculty Empowerment Pilot Project (no. CCNUAI&FE2022-03-01), the Collaborative Innovation Center for Informatization and Balanced Development of K-12 Education by MOE and Hubei Province (no. xtzd2021-005), and by the Natural Science Foundation of Hubei Province (no. 2022CFA007).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Bayesian Knowledge Tracing	BKT
Bidirectional Long Short-Term Memory	Bi-LSTM
Convolutional Neural Networks	CNN
Deep Knowledge Tracing	DKT
Graph Convolutional Networks	GCN
Graph Neural Networks	GNN
Hidden Markov Models	HMM
Item Response Theory	IRT
Learner–Resource Response	LRR
Long Short-Term Memory	LSTM
Multiple Learning Features enhanced Knowledge Tracing	MLFKT
Performance Factor Analysis	PFA
Recurrent Neural Networks	RNN
Self Attentive Knowledge Tracing	SAKT
Stacked Auto-Encoders	SAEs

References

1. Sá , M.J.; Serpa, S.; Ferreira, C.M. Citizen Science in the Promotion of Sustainability: The Importance of Smart Education for Smart Societies. *Sustainability* **2022**, *14*, 9356. [CrossRef]
2. Doukanari, E.; Ktoridou, D.; Efthymiou, L.; Epaminonda, E. The Quest for Sustainable Teaching Praxis: Opportunities and Challenges of Multidisciplinary and Multicultural Teamwork. *Sustainability* **2021**, *13*, 7210. [CrossRef]
3. Wang, Z.; Yan, W.; Zeng, C.; Tian, Y.; Dong, S. A Unified Interpretable Intelligent Learning Diagnosis Framework for Learning Performance Prediction in Intelligent Tutoring Systems. *Int. J. Intell. Syst.* **2023**, *2023*, 4468025 . [CrossRef]
4. Singh, H.; Miah, S.J. Smart Education Literature: A Theoretical Analysis. *Educ. Inf. Technol.* **2020**, *25*, 3299–3328. [CrossRef]
5. Li, L.; Wang, Z. Calibrated Q-Matrix-Enhanced Deep Knowledge Tracing with Relational Attention Mechanism. *Appl. Sci.* **2023**, *13*, 2541. [CrossRef]
6. Corbett, A.T.; Anderson, J.R. Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. *User Model. User-Adapt. Interact.* **1995**, *4*, 253–278. [CrossRef]
7. Pavlik, P.I.; Cen, H.; Koedinger, K.R. *Performance Factors Analysis—A New Alternative to Knowledge Tracing*; Technical Report; ERIC: Brighton, UK, 2009.

8. Giles, C.L.; Kuhn, G.M.; Williams, R.J. Dynamic Recurrent Neural Networks: Theory and Applications. *IEEE Trans. Neural Netw.* **1994**, *5*, 153–156. [CrossRef]
9. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.; Sohl-Dickstein, J. Deep Knowledge Tracing. In Proceedings of the NIPS'15 28th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 7–12 December 2015; Volume 1; pp. 505–513.
10. Hambleton, R.K.; Swaminathan, H. *Item Response Theory: Principles and Applications*; Springer Science & Business Media: Cham, Switzerland, 2013.
11. Schuster, M.; Paliwal, K. Bidirectional Recurrent Neural Networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
12. Abdelrahman, G.; Wang, Q.; Nunes, B. Knowledge Tracing: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
13. Song, X.; Li, J.; Cai, T.; Yang, S.; Yang, T.; Liu, C. A Survey on Deep Learning Based Knowledge Tracing. *Knowl.-Based Syst.* **2022**, *258*, 110036. [CrossRef]
14. Zhao, W.; Xia, J.; Jiang, X.; He, T. A Novel Framework for Deep Knowledge Tracing via Gating-Controlled Forgetting and Learning Mechanisms. *Inf. Process. Manag.* **2023**, *60*, 103114. [CrossRef]
15. Liu, F.; Hu, X.; Bu, C.; Yu, K. Fuzzy Bayesian Knowledge Tracing. *IEEE Trans. Fuzzy Syst.* **2022**, *30*, 2412–2425. [CrossRef]
16. Rabiner, L.; Juang, B. An Introduction to Hidden Markov Models. *IEEE ASSP Mag.* **1986**, *3*, 4–16. [CrossRef]
17. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
18. Wang, Z.; Wang, Z.; Zeng, C.; Yu, Y.; Wan, X. High-Quality Image Compressed Sensing and Reconstruction with Multi-Scale Dilated Convolutional Neural Network. *Circuits Syst. Signal Process.* **2023**, *42*, 1593–1616. . [CrossRef]
19. Zeng, C.; Zhu, D.; Wang, Z.; Wu, M.; Xiong, W.; Zhao, N. Spatial and Temporal Learning Representation for End-to-End Recording Device Identification. *EURASIP J. Adv. Signal Process.* **2021**, *2021*, 41. [CrossRef]
20. Wang, Z.; Yang, Y.; Zeng, C.; Kong, S.; Feng, S.; Zhao, N. Shallow and Deep Feature Fusion for Digital Audio Tampering Detection. *EURASIP J. Adv. Signal Process.* **2022**, *2022*, 69. [CrossRef]
21. Lyu, L.; Wang, Z.; Yun, H.; Yang, Z.; Li, Y. Deep Knowledge Tracing Based on Spatial and Temporal Representation Learning for Learning Performance Prediction. *Appl. Sci.* **2022**, *12*, 7188. [CrossRef]
22. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
23. Su, W.; Jiang, F.; Shi, C.; Wu, D.; Liu, L.; Li, S.; Yuan, Y.; Shi, J. An XGBoost-Based Knowledge Tracing Model. *Int. J. Comput. Intell. Syst.* **2023**, *16*, 13. [CrossRef]
24. Jiang, H.; Xiao, B.; Luo, Y.; Ma, J. A Self-Attentive Model for Tracing Knowledge and Engagement in Parallel. *Pattern Recognit. Lett.* **2023**, *165*, 25–32. [CrossRef]
25. Li, Z.; Yu, S.; Lu, Y.; Chen, P. Plastic Gating Network: Adapting to Personal Development and Individual Differences in Knowledge Tracing. *Inf. Sci.* **2023**, *624*, 761–776. [CrossRef]
26. Xiao, Y.; Xiao, R.; Huang, N.; Hu, Y.; Li, H.; Sun, B. Knowledge Tracing Based on Multi-Feature Fusion. *Neural Comput. Appl.* **2022**, *35*, 1819–1833. [CrossRef]
27. Wu, Z.; Huang, L.; Huang, Q.; Huang, C.; Tang, Y. SGKT: Session Graph-Based Knowledge Tracing for Student Performance Prediction. *Expert Syst. Appl.* **2022**, *206*, 117681. [CrossRef]
28. Wang, W.; Ma, H.; Zhao, Y.; Li, Z.; He, X. Tracking Knowledge Proficiency of Students with Calibrated Q-Matrix. *Expert Syst. Appl.* **2022**, *192*, 116454. [CrossRef]
29. Yeung, C.K.; Yeung, D.Y. Addressing Two Problems in Deep Knowledge Tracing via Prediction-Consistent Regularization. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale, London, UK, 26–28 June 2018; pp. 1–10. [CrossRef]
30. 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, Republic and Canton of Geneva, Perth, Australia, 3–7 April 2017; pp. 765–774. [CrossRef]
31. Sun, X.; Zhao, X.; Li, B.; Ma, Y.; Sutcliffe, R.; Feng, J. Dynamic Key-Value Memory Networks With Rich Features for Knowledge Tracing. *IEEE Trans. Cybern.* **2021**, *52*, 8239–8245. [CrossRef]
32. Liu, S.; Yu, J.; Li, Q.; Liang, R.; Zhang, Y.; Shen, X.; Sun, J. Ability Boosted Knowledge Tracing. *Inf. Sci.* **2022**, *596*, 567–587. [CrossRef]
33. Wang, W.; Ma, H.; Zhao, Y.; Yang, F.; Chang, L. SEEP: Semantic-Enhanced Question Embeddings Pre-Training for Improving Knowledge Tracing. *Inf. Sci.* **2022**, *614*, 153–169. [CrossRef]
34. Song, X.; Li, J.; Lei, Q.; Zhao, W.; Chen, Y.; Mian, A. Bi-CLKT: Bi-Graph Contrastive Learning Based Knowledge Tracing. *Knowl.-Based Syst.* **2022**, *241*, 108274. [CrossRef]
35. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
36. Wang, W.; Liu, T.; Chang, L.; Gu, T.; Zhao, X. Convolutional Recurrent Neural Networks for Knowledge Tracing. In Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chongqing, China, 29–30 October 2020; pp. 287–290. [CrossRef]
37. Pandey, S.; Karypis, G. A Self-Attentive Model for Knowledge Tracing. *arXiv* **2019**, arXiv:1907.06837.
38. Zhang, S.; Tong, H.; Xu, J.; Maciejewski, R. Graph Convolutional Networks: A Comprehensive Review. *Comput. Soc. Netw.* **2019**, *6*, 11. [CrossRef]

39. 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*; Hutter, F., Kersting, K., Lijffijt, J., Valera, I., Eds.; Springer: Cham, Switzerland, 2021; pp. 299–315. . [CrossRef]
40. Zhou, J.; Cui, G.; Hu, S.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; Sun, M. Graph Neural Networks: A Review of Methods and Applications. *AI Open* **2020**, *1*, 57–81. [CrossRef]
41. Nakagawa, H.; Iwasawa, Y.; Matsuo, Y. Graph-Based Knowledge Tracing: Modeling Student Proficiency Using Graph Neural Network. In Proceedings of the IEEE/WIC/ACM WI '19 International Conference on Web Intelligence, New York, NY, USA, 17–20 October 2019; pp. 156–163. [CrossRef]
42. Yu, M.; Li, F.; Liu, H.; Zhang, T.; Yu, G. ContextKT: A Context-Based Method for Knowledge Tracing. *Appl. Sci.* **2022**, *12*, 8822. [CrossRef]
43. Liu, D.; Zhang, Y.; Zhang, J.; Li, Q.; Zhang, C.; Yin, Y. Multiple Features Fusion Attention Mechanism Enhanced Deep Knowledge Tracing for Student Performance Prediction. *IEEE Access* **2020**, *8*, 194894–194903. [CrossRef]
44. Lu, Y.; Chen, P.; Pian, Y.; Zheng, V.W. CMKT: Concept Map Driven Knowledge Tracing. *IEEE Trans. Learn. Technol.* **2022**, *15*, 467–480. [CrossRef]
45. Wang, X.; Zheng, Z.; Zhu, J.; Yu, W. What Is Wrong with Deep Knowledge Tracing? Attention-Based Knowledge Tracing. *Appl. Intell.* **2022**, *53*, 2850–2861 . [CrossRef]
46. Long, T.; Qin, J.; Shen, J.; Zhang, W.; Xia, W.; Tang, R.; He, X.; Yu, Y. Improving Knowledge Tracing with Collaborative Information. In Proceedings of the WSDM '22 Fifteenth ACM International Conference on Web Search and Data Mining, New York, NY, USA, 21–25 February 2022; pp. 599–607. [CrossRef]
47. He, L.; Tang, J.; Li, X.; Wang, P.; Chen, F.; Wang, T. Multi-Type Factors Representation Learning for Deep Learning-Based Knowledge Tracing. *World Wide Web* **2022**, *25*, 1343–1372. [CrossRef]
48. Zhang, L.; Xiong, X.; Zhao, S.; Botelho, A.; Heffernan, N.T. Incorporating Rich Features into Deep Knowledge Tracing. In Proceedings of the L@S '17 Fourth (2017) ACM Conference on Learning @ Scale, New York, NY, USA, 20–21 April 2017; pp. 169–172. . [CrossRef]
49. Cheung, L.P.; Yang, H. Heterogeneous Features Integration in Deep Knowledge Tracing. In *Proceedings of the Neural Information Processing*; Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S.M., Eds.; Springer: Cham, Switzerland, 2017; pp. 653–662.
50. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the KDD '16 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 20–23 August 2016; pp. 785–794. [CrossRef]
51. Abdi, H.; Williams, L.J. Principal Component Analysis. *WIREs Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]
52. Zhang, N.; Du, Y.; Deng, K.; Li, L.; Shen, J.; Sun, G. Attention-Based Knowledge Tracing with Heterogeneous Information Network Embedding. In *Proceedings of the Knowledge Science, Engineering and Management*; Li, G., Shen, H.T., Yuan, Y., Wang, X., Liu, H., Zhao, X., Eds.; Springer: Cham, Switzerland, 2020; pp. 95–103.
53. Mubarak, A.A.; Cao, H.; Zhang, W. Prediction of Students' Early Dropout Based on Their Interaction Logs in Online Learning Environment. *Interact. Learn. Environ.* **2022**, *30*, 1414–1433. [CrossRef]
54. Wei, L.; Li, B.; Li, Y.; Zhu, Y. Time Interval Aware Self-Attention Approach for Knowledge Tracing. *Comput. Electr. Eng.* **2022**, *102*, 108179. [CrossRef]
55. Stamper, J.; Pardos, Z.A. The 2010 KDD Cup Competition Dataset: Engaging the Machine Learning Community in Predictive Learning Analytics. *J. Learn. Anal.* **2016**, *3*, 312–316. [CrossRef]
56. Waterworth, C.J. Relearning the Learning Curve: A Review of the Derivation and Applications of Learning-Curve Theory. *Proj. Manag. J.* **2000**, *31*, 24–31. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.