applied sciences

MDPI

Article

Bridging the Vocabulary Gap: Using Side Information for Deep Knowledge Tracing

Haoxin Xu [1], Jiaqi Yin [1], Changyong Qi [1], Xiaoqing Gu [2], Bo Jiang [1] and Longwei Zheng [3,4,*]

[1] Shanghai Institute of AI for Education, East China Normal University, Shanghai 200063, China; haoxin.xu@stu.ecnu.edu.cn (H.X.); 52215901018@stu.ecnu.edu.cn (J.Y.); changyongqi@stu.ecnu.edu.cn (C.Q.); bjiang@deit.ecnu.edu.cn (B.J.)
[2] Department of Education Information Technology, East China Normal University, Shanghai 200063, China; xqgu@ses.ecnu.edu.cn
[3] School of Education, City University of Macau, Macau 999078, China
[4] State Key Laboratory of Cognitive Intelligence, Hefei 230088, China
* Correspondence: lwzheng@cityu.edu.mo

Abstract: Knowledge tracing is a crucial task in personalized learning that models student mastery based on historical data to predict future performance. Currently, deep learning models in knowledge tracing predominantly use one-hot encodings of question, knowledge, and student IDs, showing promising results. However, they face a significant limitation: a vocabulary gap that impedes the processing of new IDs not seen during training. To address this, our paper introduces a novel method that incorporates aggregated features, termed 'side information', that captures essential attributes such as student ability, knowledge mastery, and question difficulty. Our approach utilizes side information to bridge the vocabulary gap caused by ID-based one-hot encoding in traditional models. This enables the model, once trained on one dataset, to generalize and make predictions on new datasets with unfamiliar students, knowledge, or questions without the need for retraining. This innovation effectively bridges the vocabulary gap, reduces the dependency on specific data representations, and improves the overall performance of the model. Experimental evaluations on five distinct datasets show that our proposed model consistently outperforms baseline models, using fewer parameters and demonstrating seamless adaptability to new contexts. Additionally, ablation studies highlight that including side information, especially regarding students and questions, significantly improves knowledge tracing effectiveness. In summary, our approach not only resolves the vocabulary gap challenge but also offers a more robust and superior solution across varied datasets.

Keywords: knowledge tracing; vocabulary gap; side information; deep knowledge tracing

Citation: Xu, H.; Yin, J.; Qi, C.; Gu, X.; Jiang, B.; Zheng, L. Bridging the Vocabulary Gap: Using Side Information for Deep Knowledge Tracing. *Appl. Sci.* **2024**, *14*, 8927. https://doi.org/10.3390/app14198927

Academic Editor: Douglas O'Shaughnessy

Received: 5 September 2024
Revised: 27 September 2024
Accepted: 1 October 2024
Published: 3 October 2024

1. Introduction

In the education domain, knowledge tracing (KT) is one of the basic tasks of personalized learning [1,2]. It models the state of mastery of knowledge of students based on their historical data of learning activities and predicts whether students may answer the next question correctly or not [3]. As shown in Figure 1, in the field of machine learning, KT can be viewed as a supervised learning task that takes input from students' past interactions $X = \{x_1, x_2, x_3, \ldots, x_t\}$, and predicts the future interactions x_{t+1} [4]. The record of the student practicing at time t is represented as an interaction $x_t = (e_t, r_t)$, $e_t = (s_t, k_t, q_t)$ includes the information of the students, knowledge, and questions (SKQ), and $r_t \epsilon \{0, 1\}$ represents the correctness of the answer [1]. KT aims to predict the probability that the student will give a correct response to the next question; this is represented as $P(r_{t+1} = 1 \mid e_{t+1}, X)$ [5].

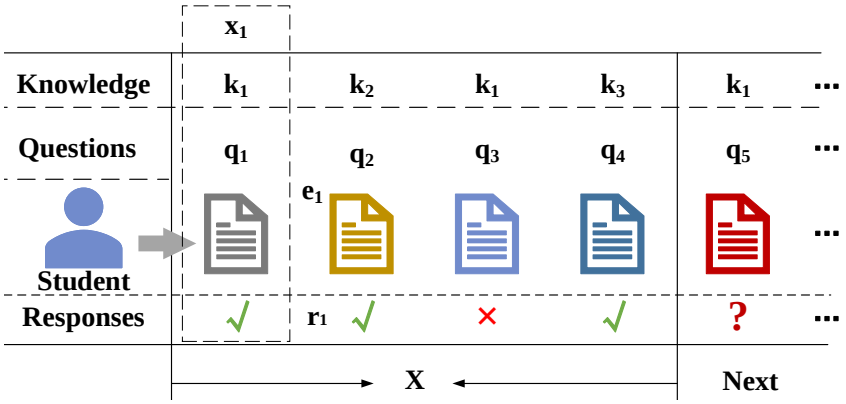

Figure 1. A toy example of knowledge tracing, where X represents the set of a student's historical interactions, x_t denotes the interaction record of the student at time t, e_t signifies the information regarding the student, knowledge, and questions at time t, and r_t indicates the correctness of the student's answers at time t.

Driven by the rapid development of deep learning technology and its powerful representation capabilities, a series of deep learning-based knowledge tracing (DLKT) models represented by deep knowledge tracing (DKT) [1,6,7], and Dynamic Key–Value Memory Networks (DKVMNs) [8] have been proposed. These DLKT models use at least one or more identification (ID) information about questions, knowledge, and students, and use this ID information as input to the model through one-hot encoding [9]. For example, a question is represented as a vector of length l, and l is the total number of questions. If the ID of the question is 10, then the value of the corresponding position in the one-hot vector will be 1. To process the sparse data, the one-hot encoding data will be reduced to low-dimensional representation through embedding [10]. Then, deep neural algorithms such as recurrent neural networks [11], attention mechanisms [12], or enhanced memory [8] are used to predict students' answers. This method allows the model to identify each SKQ so that it can achieve better performance [1,13].

However, training models with such a DLKT method will lead to insufficient generalization capability for the models [14]. Because the IDs and totals of the SKQ on different datasets are different, it is difficult for the model trained on dataset A to directly predict dataset B. This means the original model will be difficult to adapt to the new dataset, and it needs to be retrained even on the same dataset when a new student, knowledge, or question appears [1,15]. This phenomenon is called the out-of-vocabulary phenomenon, termed the vocabulary gap in this study, indicating the inability of a model to effectively process unseen vocabulary or identifiers [16]. In such cases, the one-hot encoding of question IDs may only be effective for questions present in the training dataset, rendering it unable to provide useful information for new question IDs. This limitation hinders the model's ability to generalize correctly to questions in a new dataset.

The cause of this issue lies in the excessive dependence of the model on data representation, overlooking a fundamental consideration of the essence of KT [17], specifically, what features are truly necessary for this task [18]. The hypothesis proposed in this study is that by knowing features such as the individual capability of a student, the student's mastery of knowledge, the difficulty of a question, and the complexity of the knowledge involved, one can determine whether a student can answer the next question correctly [19,20]. For instance, if a student has exhibited poor performance in their past assessments and the question they need to answer is difficult and involves complex knowledge, we can fairly accurately predict that the student may not be able to answer the question correctly. These aggregated features related to SKQ can be obtained through statistical analysis of historical answering records, termed as side information in this study, such as correctness rate and average response time. Based on this hypothesis, models trained using side information can directly predict on new datasets. The key contributions of this work can be summarized

as follows: (1) The introduction of side information: This study proposes to incorporate aggregated features related to SKQ, such as student ability, knowledge mastery, question difficulty, and knowledge complexity, as side information into the model to address the problem of over-reliance on data representation using DLKT models. (2) Solving the vocabulary gap problem: Since side information can reflect the essential characteristics of SKQ, the model trained with side information can effectively adapt to new datasets even when it encounters new students, new knowledge, or new questions, without the need for retraining, thus effectively solving the vocabulary gap problem in traditional models. (3) Improving model performance: The proposed model based on side information outperforms four baseline models on five datasets, and uses the least number of parameters. Side information of students and questions has a significant impact on the effectiveness of KT.

Overall, the main contribution of this study is the proposal to incorporate side information as input features for KT models, effectively addressing the vocabulary gap problem in DLKT models and significantly improving model performance. This approach offers a novel methodology for the field of knowledge tracing, enhancing the adaptability of models to new datasets without retraining. Furthermore, it contributes to the advancement of intelligent education by providing more accurate predictions of student performance and tailored learning experiences, ultimately fostering better educational outcomes.

2. Related Work

2.1. Deep Learning-Based Knowledge Tracing

At present, DLKT models have received considerable attention from scholars [1,6]. These models have achieved significant predictive performance compared to traditional knowledge tracing models such as Bayesian knowledge tracing [21], item response theory [22], and performance factor analysis [23]. This is due to the powerful feature extraction and function fitting capabilities of deep learning. On the one hand, several innovations have been made in the structure of knowledge tracing models. For example, the Dynamic Key–Value Memory Network (DKVMN) model utilizes dynamic key–value pairs and enhanced memory networks to better capture student knowledge states [8]. The Self-Attentive Knowledge Tracing (SAKT) model leverages the self-attention mechanism to improve the model's ability to focus on relevant past interactions [24]. Additionally, the Separated Self-Attentive Neural Knowledge Tracing (SAINT) model adopts a Transformer-based architecture [25], where exercises are embedded in the encoder and student responses are predicted in the decoder, further advancing the capability of knowledge tracing models to handle complex learning sequences.

On the other hand, mining and integrating learning-related factors is also a major trend adopted to improve the performance of DKT models. For example, scholars proposed an extensible representation learning approach for DKT to extract and integrate representations of these four types of factors, including exercise and skill, the attributes of exercise, the historical performance of the learners, and the forgetting behavior of the learners in the learning process [26]. Furthermore, a new method of contrastive deep knowledge tracing was proposed [27]. This model adopts the items with the same concepts as positive samples and the items with different concepts as negative samples, and it obtains new features from students as the input for long short-term memory networks. The models above use SKQ ID information and generate a one-hot coding vector of corresponding length according to its length, which limits the use of this model in a real environment [13]. For example, when a new SKQ appears, the original one-hot coding will no longer be applicable to the new SKQ, so the original KT model needs to be retrained. This is what this study refers to as the vocabulary gap in KT, and it is also an urgent problem that needs to be solved in the current field of KT [1].

To address the problem of the vocabulary gap, this study proposes a new methodological approach. We advocate for a focused effort on extracting shared feature representations across various datasets. These shared features, known as side information, include but are not limited to elements such as question difficulty, student ability, and the complexity

of knowledge, and do not involve any specific ID information. By this means, we can capture more universal and essential data characteristics, thereby facilitating the effective transfer and application of the DLKT model in new and unknown datasets. Additionally, this method aims to enhance the generalizability of the model, enabling it to be more robust and effective across different educational environments and scenarios.

2.2. Knowledge Tracing with Side Information

In the realm of DLKT models, recent studies have made significant strides by incorporating diverse types of side information to enhance model performance [28]. In particular, models such as deep knowledge tracing with side information have employed graph embedding algorithms to integrate question–knowledge relationships, thus enriching the modeling of student knowledge status [29]. Additionally, other approaches like the exercise-aware knowledge tracing and DKT-forget models have leveraged question text and forgetting-related data, respectively, refining their predictive capabilities [30,31]. Furthermore, advanced models, including leveled attentive knowledge tracing [32] and relation-aware self-attention models [33,34], have adopted hierarchical and contextual learning strategies, grouped students, and built contextual information from text content, performance data, and forgetting information.

However, this reliance on additional data sources, such as question IDs, knowledge structure, question text, and students' personal information, presents notable challenges. It especially limits the applicability of these models to new datasets where such side information may not be readily available. These observations suggest a future research trajectory toward developing methods that can effectively utilize available data or creating models that are less dependent on extensive side information, aiming to broaden their applicability across various educational settings.

3. Side Information of Knowledge Tracing

3.1. Knowledge Tracing Framework

In view of several major problems in current DLKT models, that is, as the use of ID information makes it difficult for the model to adapt to the newly generated data and side information is difficult to obtain, we propose a new KT framework base on side information. As is widely acknowledged, students, knowledge, and questions predominantly engage in the KT aspect of intelligent tutoring systems. In this context, a student's mastery of a specific knowledge component is gauged through their responses to a series of related questions. The overarching objective of KT is to encapsulate comprehensive information—including the student's mastery of knowledge, question difficulty levels, and the inherent complexity of these knowledge—up to the time point $t+1$. Furthermore, it aims to develop a predictive model that estimates the likelihood of a student correctly answering subsequent questions.

Whether a student can answer a question q correctly at time $t+1$ or not is jointly determined by the characteristics of the three elements of the SKQ at time $t+1$, and this process of the decision can be fitted with a function or model to predict whether the student can answer the question correctly at time $t+1$. We can obtain $y = f(S_{t+1}, K_{t+1}, Q_{t+1})$. However, information such as students' mastery of knowledge S_{t+1}, the complexity of knowledge K_{t+1}, and the difficulty of questions Q_{t+1} are implicit attributes of the above three elements, and their ground truth values cannot be directly obtained. So, they can only be represented through existing explicit information such as interaction records. For example, the correct rate of students' answers to the same knowledge is used to indicate the student's knowledge mastery, and the overall correct rate of the question is used to indicate the difficulty of the question. The implicit information of SKQ is represented through their related historical observable data. This represents information that can be used to support the prediction of the correct rate of students' future answers.

The implicit information of SKQ at time $t+1$ can be characterized by the acquisition and analysis of explicit data before time $t+1$, where x_t represents the record obtained by the system at time t. We can obtain $S_{t+1} = g_s(x_1, x_2, \ldots, x_t)$, $Q_{t+1} = g_q(x_1, x_2, \ldots, x_t)$, and

$K_{t+1} = g_k(x_1, x_2, \ldots, x_t)$. Implicit information is an attribute possessed by each SKQ. If the ID information that can identify these entities is removed, this implicit information can be regarded as side information because its use as model input is not limited by the ID or the number of entities. So, we propose a new KT framework based on side information; the model constructed based on the side information obtained by training on dataset A can be directly applied to dataset B and can achieve the same performance. In addition, the model will become more general as it was trained on more datasets, potentially solving the problem of catastrophic forgetting in continual learning.

3.2. Side Information Extraction

In normal circumstances, student interaction records are generally stored in learning systems, and every interaction record includes six fields, as shown in Table 1. Considering that the original data only contain six dimensions and that the features input into the model cannot include all the ID information, we propose 14 features as side information about SKQ for each interaction record (Table 2). The selection of these side information is not accidental. These are the features related to SKQ that can be extracted from the known data. They determine the right and wrong answers of students to a certain extent.

Table 1. Fields of interactive records in the dataset.

Field	Description	Example
stu_id	Student ID.	23
q_id	Question ID.	141
k_id	Knowledge ID.	212
use_time	Answer time.	60 s
is_right	Correct or incorrect.	0 or 1
ts	Timestamp.	2009-08-13 14:30:12

Table 2. All side information about students, questions, and knowledge components can be derived from six known fields.

SKQ	Dimension	Description	Name	Index
Student	Mastery	S_1's correct rate for K_1.	*sk_acc*	1
		Average time spent by S_1 on K_1.	*sk_mt*	2
		The number questions answered by S_1 on K_1.	*sk_n*	3
		The elapsed time since the last record of K_1.	*sk_lt*	4
	Ability	S_1's correct rate for all questions.	*sq_acc*	5
		The average time spent by S_1 on all questions.	*sq_mt*	6
		The number questions answered by S_1.	*sq_n*	7
		The elapsed time since the last record.	*sq_lt*	8
Knowledge	Complexity	The correct rate of K_1.	*k_acc*	9
		The average time of K_1.	*k_mt*	10
		The number of K_1 records.	*k_n*	11
Question	Difficulty	The correct rate of Q_1.	*q_acc*	12
		The average time of Q_1.	*q_mt*	13
		The number of Q_1 records.	*q_n*	14

For example, at time $t+1$, we have dataset $X = \{x_1, x_2, x_3, \ldots, x_t\}$, and the interaction record x_{t+1} is used to predict; if its $stu\_id = S_1, q\_id = Q_1$, and $k\_id = K_1$, its side information is obtained as follow:

(1) Obtain all the records of student S_1 about knowledge K_1 from X, and calculate the mean value of *is_right* (*sk_acc*), the mean value of *use_time* (*sk_mt*), the length of the records (*sk_n*), and the elapsed time since the last record (*sk_lt*). These four features are about the mastery of knowledge K_1 for student S_1. If *sk_acc* is higher, *sk_mt* is less, *sk_n* is more, and *sk_lt* is shorter, this means that S_1 may have mastered K_1 at the current moment.

(2) Obtain all the records of student S_1 from X, and calculate the mean value of *is_right* (*sq_acc*), the mean value of *use_time* (*sq_mt*), the length of the records (*sq_n*), and the elapsed time since the last record (*sq_lt*). These four features are about the overall ability of S_1. If *sq_acc* is high, *sq_mt* is less, *sq_n* is more, and *sq_lt* is shorter, it means that the student has a strong learning ability and has a good grasp of all the knowledge components learned.

(3) Obtain all the records of knowledge K_1 from X, and calculate the mean value of *is_right* (*k_acc*), the mean value of *use_time* (*k_mt*), and the length of the records (*k_n*). These three features are about the complexity of K_1. If *k_acc* is low, *k_mt* is large, and *k_n* is large, it means that most students do not have a good grasp of K_1, and it also means that K_1 is complex and difficult to master.

(4) Obtain all the records of question Q_1 from X, and calculate the mean value of *is_right* (*q_acc*), the mean value of *use_time* (*q_mt*), and the length of the records (*q_n*). These three features are about the difficulty of Q_1. If *q_acc* is low, *q_mt* is large, and *q_n* is large, it means that most students do not answer Q_1 correctly and it takes a long time, which also means that Q_1 is difficult. Here, the number of answers and the time from the last answer are used as adjustment items for the correct rate and the average time. Even if the correct rate is 100, if there is only one answer record and several months have passed, the reliability of the correct rate will be greatly reduced. Therefore, it is also necessary to input the model as a feature and let the model learn this relationship.

4. Knowledge Tracing Model Based on Side Information

Based on the KT framework and side information mentioned above, we built a KT model based on side information: SIKT (see Figure 2). The input of SIKT is divided into two parts. The first part is a vector x_{t+1} (see Formula (1)) about the side information of the interaction record, and its length is 14. The second part is two sequences of data, including the student's five knowledge-related interaction records before time $t+1$, $SK_{t+1} = (x_{sk\_an_1}, x_{sk\_an_2}, \ldots, x_{sk\_an_5})$, and the student's five interaction records before time $t+1$, $SQ_{t+1} = (x_{sq\_an_1}, x_{sq\_an_2}, \ldots, x_{sq\_an_5})$. The length of each vector in SK_{t+1} and SQ_{t+1} is 16, consisting of 14 side information features and 2 additional pieces of information (*is_right*, *use_time*). If the length of the historical records is less than five, fill the remaining vector with all 0 s. As shown in Formula (2), $x_{sq\_an_1}$ contains 16 features, and the data are obtained at time $t-4$.

$$x_{t+1} = (sk\_acc, sk\_mt, sk\_n, sk\_lt, sq\_acc, sq\_mt, sq\_n, sq\_lt, k\_acc, k\_mt, k\_n, q\_acc, q\_mt, q\_n)_{t+1} \quad (1)$$

$$x_{sk\_an_1} = (sk\_acc, sk\_mt, sk\_n, sk\_lt, sq\_acc, sq\_mt, sq\_n, sq\_lt, k\_acc, k\_mt, k\_n, q\_acc, q\_mt, q\_n, is\_right, use\_time)_{t-4} \quad (2)$$

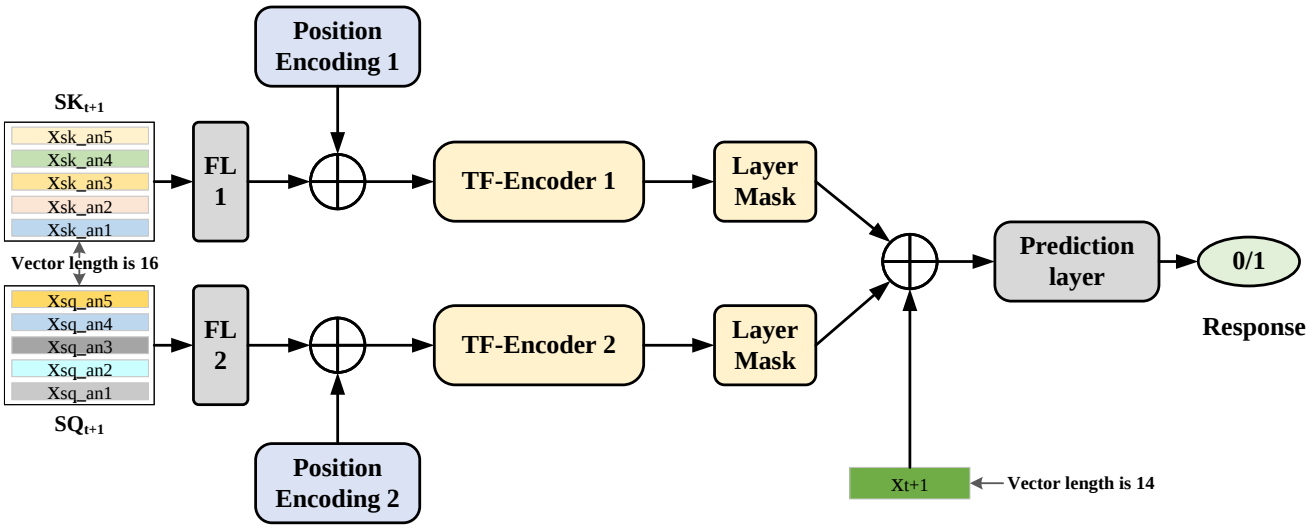

Figure 2. Framework of SIKT model.

4.1. Encoder Layer

The encoder layer in our study is an encoder module of the standard Transformer model [35], which is mainly used to process student historical data. Firstly, the two sequences of data (SK_{t+1} and SQ_{t+1}) pass through a fully connected layer (see Figure 2 FL1 and FL2) to expand the data to the specific dimension (see Formulas (3) and (4)), and then the position information (see Figure 2 Position Encoding 1 and Position Encoding 2) learned through an embedding layer is spliced [24]. Finally, the sequences of data enter the two TF-Encoder layers, respectively (see Formulas (5) and (6)).

$$SK_{fl} = SK_{t+1}W_1 + b_2 \tag{3}$$

$$SQ_{fl} = QK_{t+1}W_2 + b_2 \tag{4}$$

$$SK_{tf} = Encoder(SK_{fl} + Position(SK_{fl})) \tag{5}$$

$$SQ_{tf} = Encoder(SQ_{fl} + Position(SQ_{fl})) \tag{6}$$

These two TF-Encoder layers adopt the same structure, as shown in Figure 3. They are encoder modules within the Transformer architecture, utilizing their multi-head attention mechanism to process time-series data and extract features [35]. Each encoder consists of multiple layers, each comprising multi-head self-attention and a feed-forward network. The multi-head attention mechanism allows the model to attend to different positions of the input sequence simultaneously. The formula is as follows:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O \tag{7}$$

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{8}$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{9}$$

$$\text{output} = \text{LayerNorm}(\text{input} + \text{Sublayer}(\text{input})) \tag{10}$$

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{11}$$

Here, Q, K, V represent the query, key, and value matrices, respectively; W^Q, W^K, W^V, W^O are learnable parameter matrices; and d_k is the dimension of the key vectors used for scaling the dot product. The output of each submodule (self-attention and feed-forward network) is followed by a residual connection and then layer normalization (see Formula (10)), where Sublayer(input) is the output of the sub-layer itself. Each encoder layer also includes a feed-forward network, which is a series of linear transformations applied independently but identically at each position (see Formula (11)), where W_1, W_2, b_1, b_2 are the parameters of the feed-forward layers.

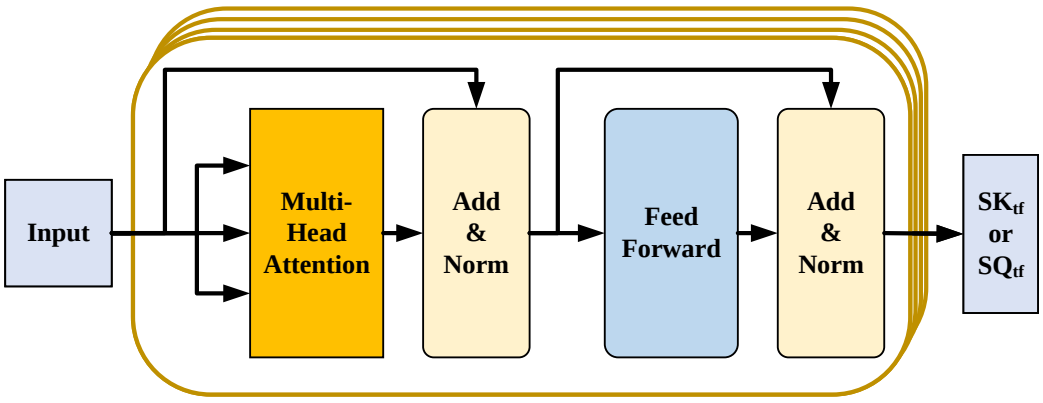

Figure 3. Framework of TF-Encoder layer.

4.2. Layer Mask Layer

The data output from the encoder is a sequence of data. Since the length of the original input sequence may be less than five, the filled vector needs to be ignored, and then the two processed sequences of data are accumulated and averaged along the time dimension. Eventually, the sequences of data are compressed into two vectors (sk_{t+1} and sq_{t+1}; see Formulas (12) and (13)) as a representation of the student's mastery of knowledge components and the student's overall ability.

$$sk_{t+1} = \sum_{i=1}^{I=5} x_{sk\_an_i}, x_{sk\_an_i} \neq 0, x_{sk\_an_i} \in SK_{tf} \tag{12}$$

$$sq_{t+1} = \sum_{i=1}^{I=5} x_{sq\_an_i}, x_{sq\_an_i} \neq 0, x_{sq\_an_i} \in SQ_{tf} \tag{13}$$

4.3. Prediction Layer

Before entering the prediction layer, it is also necessary to merge the data by splicing x_{t+1}, sk_{t+1}, and sq_{t+1} along the direction of the feature dimension to obtain a long vector and input it into the prediction layer (see Formula (14)). The prediction layer is composed of multiple fully connected layers, and the output data pass through the sigmoid function to output the final predictive result p_t. In Formula (14), $\bigoplus$ denotes the concatenation of two vectors, W_p represents the weights of the fully connected layer, b_p is the bias term of the fully connected layer, and *Sigmoid* indicates the sigmoid activation function. The purpose of this equation is to compute the prediction probability p_t at the current time step, which integrates the concatenated input vectors x_{t+1}, sk_{t+1}, and sq_{t+1} through the fully connected layer. If p_t is less than 0.5, the output is 0; otherwise, the output is 1.

$$p_t = Sigmoid((x_{t+1} \bigoplus sk_{t+1} \bigoplus sq_{t+1}) * W_p + b_p) \tag{14}$$

5. Experiments

5.1. Datasets

As shown in Table 3, five different real-world datasets were used in the experiments, namely ASSISTment2009 (A09) [36], ASSISTment2012 (A12) [37], Riiid's EdNet (EN) [38], Junyi Academy (Junyi) [39], and NeurIPS 2020 Education Challenge (Eedi) [40]. These five datasets are commonly used in the KT domain and all contain the six fields mentioned above [1,41]. The A09 and A12 are datasets of student interaction records collected and provided by the online tutoring system ASSISTments in 2009 and 2012 [42]. The similarities between the A09 and A12 datasets lie in the fact that both were collected using the same system. However, they differ in terms of the time periods during which the data were collected, as well as the students, questions, and knowledge involved [43]. The EdNet dataset was collected over a two-year period from the intelligent online tutoring platform Riid TUTOR, dedicated to practicing English for an international communication assessment in South Korea [44]. Due to the large amount of raw data (70 million records), we sorted the data by timestamp and extracted the top three million records as the dataset (EN) for the experiments. The Junyi dataset was compiled over a period from November 2010 to March 2015, sourced from the Junyi Academy [39], an online learning platform operating in Taiwan. The Eedi dataset includes student responses to mathematics questions collected from September 2018 to May 2020 [40]. Eedi is an educational platform used by millions of students worldwide, offering diagnostic multiple-choice questions for students aged seven to eighteen.

All datasets were randomly divided into training sets and test sets according to the students (Table 3). The training set was only used for model training, and the test set was only used to evaluate the model. This means that there was no overlap between the students in the training set and the test set. The model needed to predict the results of students who have never seen it before.

Table 3. The number of students, questions, knowledge, and interaction records in all datasets.

Dataset	Students	Questions	Knowledge	Records
A09	8093	6902	542	602,956
A09-train *	5093	6282	534	385,806
A09-test	3000	5791	536	217,150
A12	29,018	53,086	265	2,711,602
A12-train	26,018	52,697	265	2,434,816
A12-test	3000	40,040	255	276,786
EN	385,188	8339	1017	3,000,000
EN-train	335,188	8212	988	2,611,397
EN-test	50,000	6648	751	388,603
Junyi	19,323	705	39	2,879,392
Junyi-train	15,458	705	39	2,271,892
Junyi-test	3865	674	39	607,500
Eedi	20,000	27,612	282	3,354,687
Eedi-train	16,000	27,611	282	2,672,901
Eedi-test	4000	26,949	281	681,786

* A09-train refers to the training set in the A09 dataset, while A09-test refers to the testing set in the A09 dataset, with other datasets following similarly. The students in the training set and testing set are mutually exclusive.

5.2. Experimental Setup

Firstly, we used the DKT [6], DKVMN [8], SAKT [24], and SAINT [25] models as the baseline models for the testing on five datasets. The four models were used as the baseline models because both of them encode the ID of the question or knowledge in a one-hot manner and use the response sequence of the same student as the model input. Secondly, we explored whether the proposed model can be trained on a dataset and directly predicted on a new dataset. For example, we used a model trained on the A09 dataset to make predictions directly on the other four datasets. After conducting such experiments on all five datasets, we merged them together and trained the model, called joint training, and provided the results of the model on the five datasets. Finally, the model evaluation metrics used in this experiment were the Area Under the Curve (AUC) and accuracy (ACC), which are also the most commonly used metrics in the KT domain [45]. The loss function used in model training was binary cross-entropy with logit loss, the optimizer was Adam, each model was trained for 50 epochs from scratch, and the learning rate was 0.001.

6. Results

6.1. Comparison with Baseline Models

We compared the SIKT model with the DKT, DKVMN, SAKT, and SAINT models; the results are shown in Table 4. The results demonstrate that the SIKT model showed the most outstanding performance on the A09 and A12 datasets. Specifically, it achieved the highest AUC (80.73) and near-peak ACC (72.31) on the A09 dataset, while simultaneously obtaining the highest AUC (75.90) and ACC (73.20) on the A12 dataset, and the highest AUC on the Junyi (72.71) and Eedi (78.32) datasets. In comparison, although the DKT model achieved the highest ACC (72.72) on the A09 dataset, its AUC (79.87) was slightly lower than that of the SIKT model (80.73). On the EN dataset, the SAINT model exhibited the best performance, with the highest AUC (77.14) and ACC (70.13), indicating its ability to effectively capture learning patterns in this dataset.

Regarding parameter size, the SIKT model stands out with only 57.24 k parameters, yet it delivered excellent results across multiple datasets. In particular, on the A09 and A12 test sets, it achieved the highest AUC and ACC, and the highest AUC on the Junyi and Eedi datasets, showcasing its efficiency with a smaller parameter footprint. In contrast, other models such as DKVMN (291.70 k), DKT (651.62 k), SAKT (847.36 k), and SAINT (71,184.90 k) have significantly larger parameter sizes, with the SAINT model having the largest. Despite its strong performance on the EN dataset, SAINT's substantial parameter size could introduce additional computational overhead.

In summary, the SIKT model achieves a well-balanced trade-off between performance and resource utilization. It outperformed the DKT, DKVMN, and SAKT models on the A09, A12, Junyi, and Eedi datasets, while maintaining overall stability. Thanks to its significantly reduced parameter size, it was able to maintain high classification accuracy and AUC, making it particularly suitable for environments with limited computational resources. While the SAINT model showed the best performance on the EN dataset, its large parameter size may limit its practical applicability in resource-constrained scenarios.

Table 4. Results of different models on five datasets.

Models	A09-Test		A12-Test		EN-Test		Junyi-Test		Eedi-Test		*p.um*
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	
DKVMN *	70.39	65.31	67.81	70.88	65.85	60.94	69.22	88.77	71.87	68.52	291.70 k
DKT	79.87	72.72	72.82	73.12	68.67	64.67	69.99	88.83	74.45	70.04	651.62 k
SAKT	79.78	72.20	71.78	73.06	70.31	64.41	69.51	88.81	73.84	69.56	847.36 k
SAINT	75.62	69.18	65.06	70.02	77.14	70.13	70.20	88.61	63.66	62.62	71,184.90 k
SIKT	80.73	72.31	75.90	73.20	72.95	65.64	72.71	88.49	78.32	53.87	57.24 k

* The DKVMN model was trained on the A09-train dataset and evaluated on the A09-test dataset; the same strategy was applied to other models.

6.2. Results on the New Dataset

To demonstrate the vocabulary gap problem in DLKT models, we set the problem length as the sum of the problem lengths from all datasets and used the DKT and SAINT models to directly predict performance on the new combined dataset. As shown in Table 5, the DKT model performed well on the original training dataset, with DKT-A09 achieving an AUC of 79.87 on the A09 dataset. However, when these DKT models were applied to new datasets, a substantial drop in performance was observed. For instance, the DKT-A09 model exhibited significantly lower AUC values of 60.92 and 52.63 on the A12 and EN datasets, respectively. Similarly, the generalization capabilities of the DKT-A06 and DKT-EN models were limited when applied to other datasets, showing considerable performance degradation. A similar pattern was observed with the SAINT model. Although the SAINT-EN model performed best on the EN dataset (AUC: 77.14, ACC: 70.13), its performance dropped sharply on the A09 (AUC: 51.03, ACC: 51.42), A12 (AUC: 49.56, ACC: 50.54), Junyi (AUC: 50.33, ACC: 40.12) and Eedi (AUC: 49.13, ACC: 46.82) datasets. This phenomenon underscores the vocabulary gap issue in DLKT models. The vocabulary gap arises because the new datasets contain SKQ that the DLKT models have not encountered during training, resulting in poor model transfer. The models' struggle to maintain performance on datasets with unfamiliar elements highlights the need for enhancing the generalization capabilities of DLKT models across diverse datasets.

Table 5. Results of DLKT models on new datasets.

Models	A09-Test		A12-Test		EN-Test		Junyi-Test		Eedi-Test	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
DKT-A09 *	79.87	72.72	60.92	67.46	52.63	50.41	54.00	75.53	50.62	53.13
DKT-A12	65.78	61.70	72.87	73.12	48.41	49.51	56.10	78.56	52.87	61.38
DKT-EN	51.03	51.42	49.56	50.54	68.67	64.67	52.58	65.66	50.11	52.67
DKT-Junyi	51.02	52.41	52.20	59.88	46.66	47.20	69.99	77.83	51.22	53.49
DKT-Eedi	70.74	65.07	63.79	67.61	44.91	46.68	56.84	88.53	74.45	70.04
SAINT-A09	75.62	69.18	50.60	58.07	56.08	52.94	48.96	58.21	49.99	54.58
SAINT-A12	48.15	52.10	65.06	70.02	53.14	55.41	50.73	77.20	49.62	59.59
SAINT-EN	51.03	51.42	49.56	50.54	77.14	70.13	50.33	40.12	49.13	46.82
SAINT-Junyi	49.82	56.22	50.35	69.74	48.11	55.89	70.20	88.61	49.44	63.15
SAINT-Eedi	53.68	54.61	50.12	59.51	44.06	51.71	48.10	60.28	63.66	62.62

* The DKT-A09 model was trained on the A09-train dataset and evaluated on the A09-test, A12-test, EN-test, Junyi-test, and Eedi-test datasets; the same strategy was applied to other models.

Table 6 presents the results of the SIKT model trained on one dataset and predicted on another dataset. Firstly, after the model had trained on one dataset, it cloud directly predict other datasets, and achieve comparable performance, such as the SIKT-EN model obtained by training on EN (AUC: 72.95, ACC: 65.64) being able to predict on A09 (AUC: 76.65, ACC: 65.98), A12 (AUC: 71.24, ACC: 56.47) Junyi (AUC: 67.71, ACC: 71.02), and Eedi (AUC: 61.29, ACC: 58.84) directly, and the situation was similar for other datasets. This indicates that although the model has not seen the students, problems, or knowledge in the other datasets, it can still achieve good performance on the new dataset. Secondly, the AUC and ACC were similar when SIKT-A12 was tested on A09 (AUC: 79.76, ACC: 72.87) compared to SIKT-A09 tested on A09 (AUC: 80.73, ACC: 72.31). It is shown that the model can acquire more shared knowledge when learning similar tasks because these two datasets come from different years in the same learning system. Thirdly, it is also interesting to note that SIKT-A12 and SIKT-EN even showed higher test performances on A09-test than on A12-test and EN-test. This indicates that the model can perform well on a smaller number of datasets after learning a database that contains a larger number of SKQ. Therefore, we further merged all the datasets to train SIKT. Finally, it can be seen that during the joint training, the model achieved the highest performance on five datasets, as it was able to learn the feature distribution of all datasets at once and perform the best on all five datasets simultaneously. This also means that using SIKT can train data from multiple datasets simultaneously, without constantly bloating the model due to the increase in SKQ, solving the problem of the DLKT model being unable to expand quickly.

Table 6. Results of SIKT model on new datasets.

Models	A09-Test		A12-Test		EN-Test		Junyi-Test		Eedi-Test	
	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC	ACC
SIKT-A09 [1]	80.73	72.31	73.95	65.20	69.73	65.48	66.67	78.90	69.03	59.48
SIKT-A12	79.76	72.87	75.90	73.20	70.95	67.88	65.29	70.63	63.20	58.22
SIKT-EN	76.65	65.98	71.24	56.47	72.95	65.64	67.71	71.02	61.29	58.84
SIKT-Junyi	69.98	62.10	65.32	61.76	61.11	60.00	72.71	88.49	68.60	64.37
SIKT-Eedi	75.26	60.43	67.69	60.27	59.41	56.49	60.66	70.22	78.32	59.87
Join trian [2]	80.77	72.36	76.04	73.15	73.11	65.70	72.80	88.50	79.01	64.07

[1] The SIKT-A09 model was trained on the A09-train dataset and evaluated on the A09-test, A12-test, EN-test, Junyi-test, and Eedi-test datasets. The same strategy was applied to other models. [2] The model was trained by mixing the training datasets from all datasets together, and then its performance on different test sets was evaluated.

Overall, we found that SIKT can be trained on one dataset and directly predict on new datasets, while maintaining considerable performance. It can also learn all tasks at the same time and achieve acceptable performance. What this means is that if the model is allowed to learn from as many different datasets with a wide distribution of data, it obtain a more general model that can support more KT tasks.

6.3. Ablation Experiments Based on Different Features

In order to further verify the actual role of side information in the KT, we removed the side information of students, questions, and knowledge in turn, and reported the SIKT model effect of missing different side information. The AUC results on five datasets are shown in Table 7. Firstly, when removing features related to student knowledge mastery, the performance of the model on five datasets decreased by 1.18 (A09), 3.14 (A12), 0.08 (EN), 0.97 (Junyi), and 1.85 (Eedi). When removing features related to student's abilities, the performance of the model on five datasets decreased by 3.21 (A09), 2.17 (A12), 0.99 (EN), 0.09 (Junyi), and 2.89 (Eedi). When features related to student characteristics are removed, the performance of the model decreases by the largest amounts. It decreased by 6.67 (A09), 6.94 (A12), 1.00 (EN), 2.47 (Junyi), and 11.14 (Eedi) on the five datasets. This indicates that in KT, the representation of students' abilities is very important because it can largely determine whether the student can answer a certain question. Second, after removing the

side information of the question, the AUC decreased by 3.95 (A09), 6.07 (A12), 3.53 (EN), 4.17 (Junyi), and 3.86 (Eedi). This indicates that the side information of the question is also a very important feature because the difficulty of the question can also determine whether the student can answer the question correctly. Finally, it is also interesting to note that when removing knowledge-related side information, the AUC only decreased by 0.91 (A09), 1.54 (A12), 0.05 (EN), 0.37 (Junyi), and 0.09 (Eedi). It indicates that in KT, knowledge-related side information has the smallest impact on the effect. This is a counterintuitive finding; the explanation we can think of is whether KT primarily focuses on the relationship between students and questions, where knowledge plays a certain connecting role between the two, that is, questions classified with different knowledge are presented to students. In summary, it can be found that the side information related to students and questions has the greatest impact on model performance, while the side information related to knowledge has the smallest impact on model performance.

Table 7. AUC results of SIKT model when removing different side information.

Input	Remove	A09	A12	EN	Junyi	Eedi
5–14	1–4 [a]	79.55 (−1.18)	72.76 (−3.14)	72.87 (−0.08)	71.74 (−0.97)	76.47 (−1.85)
1–4, 9–14	5–8 [b]	77.52 (−3.21)	73.73 (−2.17)	71.96 (−0.99)	72.62 (−0.09)	75.43 (−2.89)
9–14	1–8 [c]	73.97 (−6.76)	68.96 (−6.94)	71.95 (−1.00)	70.24 (−2.47)	67.18 (−11.14)
1–8, 12–14	9–11 [d]	79.82 (−0.91)	74.36 (−1.54)	72.90 (−0.05)	72.34(−0.37)	78.23 (−0.09)
1–11	12–14 [e]	76.78 (−3.95)	69.83 (−6.07)	69.42 (−3.53)	68.54 (−4.17)	74.46 (−3.86)
1–14	-	80.73	75.90	72.95	72.71	78.32

[a] Removed student mastery. [b] Removed student's ability. [c] Removed student characteristics. [d] Removed knowledge complexity. [e] Removed question difficulty.

6.4. Explainability Analysis Using SHAP

To further analyze the importance of each feature, this study used Shapley values to analyze the importance and contribution of each feature in the prediction. Shapley Additive Explanations (SHAP) is a method used to explain model predictions by calculating the contribution of each feature to the model's output [46]. Figure 4 shows the importance of all features, ranked from high to low. The darker red indicates higher feature values, and the lighter blue indicates lower feature values. The horizontal axis is the SHAP value of the corresponding feature, which is the impact size of that feature on the model output. If the SHAP value is negative, it means that the feature will decrease the probability of the student answering correctly. If it is positive, it will increase the probability.

We can see that *q_acc*, *sq_acc*, *sk_acc*, and *q_mt* were the four most important features, which is also consistent with previous experimental results indicating that features about questions and students have the greatest impact on model performance. When *q_acc* has a higher value, the model tends to predict that the student can answer the question correctly, because most of the students can answer this question correctly. Similarly, when *sq_acc* or *sk_acc* are higher, the model also tends to predict that the student can answer the question correctly because the student has a stronger ability or has mastered the knowledge well enough. On the contrary, when the average time to answer a question (*q_mt*) is longer, it indicates that the question is more difficult, so the student is also less likely to answer it correctly. When the number of questions the student has answered for a knowledge (*sk_n*) increases, it means that the student has not yet mastered that knowledge fluently, so they need more practice. Therefore, the model tends to predict that the student cannot answer the question correctly.

Specifically, we used two case studies to explain how each feature affects the output of the model. As shown in Figures 5 and 6, blue represents that the feature has a negative impact on the prediction, and red represents that the feature has a positive impact on the prediction. In Case 1, the model output was 0.84, indicating a high likelihood of a correct answer. Three features significantly impacted this outcome: *sq_acc* was 70.38, *sk_acc* was 66.67, and *q_acc* was 40.51. The first two had a positive impact on the results, while the

third had a negative impact. This means that although the accuracy of this question was only 40.51, this student had stronger ability and had basically mastered this knowledge. Therefore, the model predicted that the student can answer this question correctly.

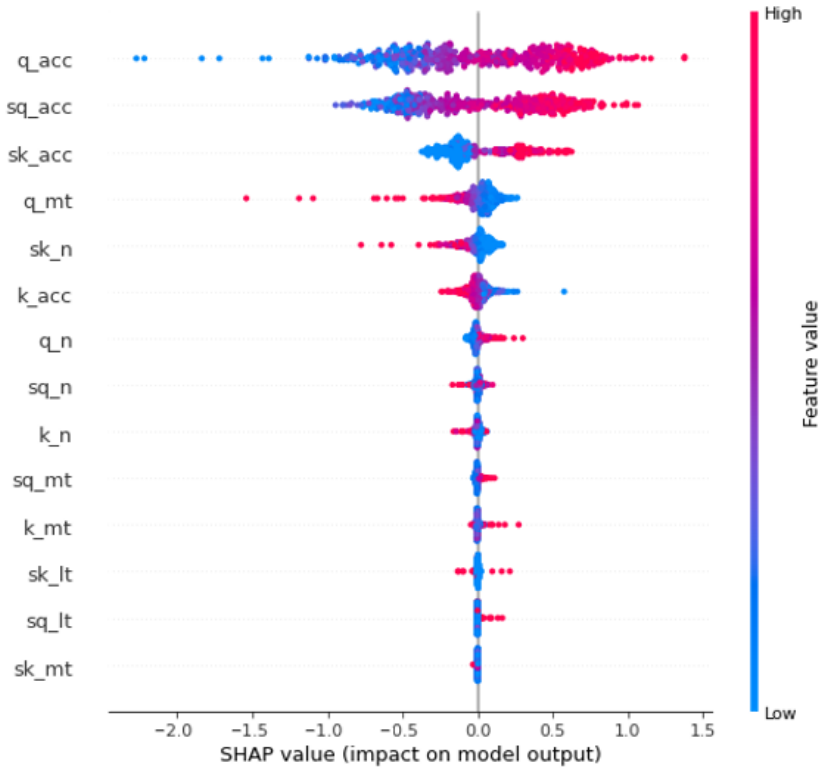

Figure 4. The impact of different feature values on the model results.

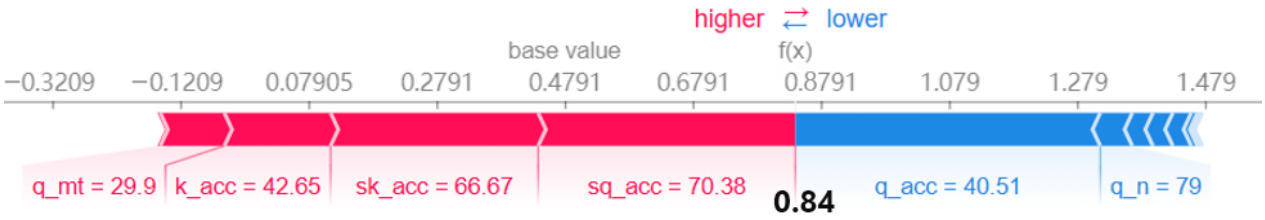

Figure 5. Interpretability analysis of Case 1. The model's inference result $f(x)$ is 0.84, and the true label is 1. Red features positively influence the model output, while blue features negatively influence it.

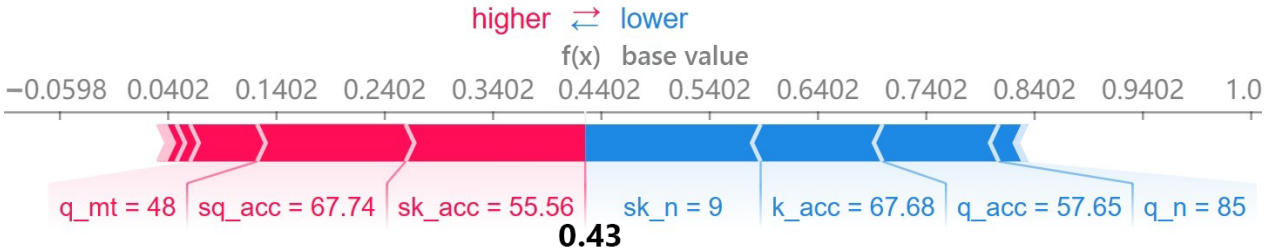

Figure 6. Interpretability analysis of Case 2. The model's inference result $f(x)$ is 0.43, and the true label is 0. Red features positively influence the model output, while blue features negatively influence it.

In Case 2, the situation was somewhat different, and the model output was 0.43. The five features that had the greatest impact on the results were *sk_acc*, *sq_acc*, *sk_n*, and *q_acc*, where the first two had a positive impact, and the last three had a negative impact. Although this student had already answered nine questions on this knowledge topic, the

student knowledge mastery level was only 56.56, and the accuracy rate of this knowledge was 67.68, with a question difficulty level of 57.65. Therefore, the model predicted that the student will not be able to correctly answer this question.

By analyzing the reasons that affect the output results of a single sample, we can analyze the reasons for each student's success or failure on each question, and provide support for personalized learning for students. This is also the advantage of using side information for KT.

7. Conclusions

In this paper, we elaborated on the vocabulary gap issue in DLKT models. To address this challenge, we proposed an innovative approach by introducing 14 aggregate features as side information, leading to the development of the SIKT model. These features are specifically crafted to capture the intricate relationships between students, questions, and knowledge components, thereby minimizing the dependency on specific ID information. This design allows the SIKT model to be trained on one dataset and seamlessly applied to another, effectively bridging the vocabulary gap. Experimental results demonstrate the significant effectiveness of the proposed approach. Furthermore, the model's remarkable cross-dataset adaptability suggests its potential for broad application in various educational platforms, enabling more personalized learning experiences without the need for extensive retraining. These findings pave the way for future research toward developing a more universal KT model by integrating diverse data sources.

Looking ahead, there are several aspects worthy of further exploration and refinement. Firstly, the extraction of side information still has room for improvement, especially in obtaining students' demographic information and learning styles [11,26,47]. Future work could explore acquiring these valuable features from alternative data sources and integrating them into the model to achieve more comprehensive and accurate student modeling. Secondly, the model structure used in this study is relatively simple, employing a basic neural network for predictions. Future research could investigate and compare different model architectures to determine whether more complex structures or those tailored to specific tasks could produce better results [48,49]. Finally, this study relied on three real-world datasets to validate the model's generalizability. To comprehensively assess the robustness and generalizability of the model, future work could involve the inclusion of more diverse datasets from different sources [1,13].

Author Contributions: Conceptualization, H.X., B.J. and X.G.; data curation, J.Y. and C.Q.; funding acquisition, L.Z. and X.G.; methodology, H.X., X.G., B.J. and L.Z.; project administration, X.G. and L.Z.; resources, J.Y.; software, C.Q.; supervision, X.G., B.J. and L.Z.; visualization, J.Y. and C.Q.; writing—original draft, H.X. and J.Y.; writing—review and editing, H.X., C.Q., B.J. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Opening Foundation of the State Key Laboratory of Cognitive Intelligence (grant number iED2023-008) and the National Natural Science Foundation of China (grant number 62477013). We express our gratitude for the financial support, which enabled the successful completion of this research.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets can be found in the following repositories: ASSISTment2009: https://drive.google.com/file/d/0B2X0QD6q79ZJbFI2ZlRBbTk1MjQ/view?resourcekey=0-B0T0KRHYM_s7E34ur3rHoQ, accessed on 1 June 2024, reference [36]; ASSISTment2012: https://drive.google.com/file/d/1cU6Ft4R3hLqA7G1rIGArVfelSZvc6RxY/view, accessed on 1 June 2024, reference [37]; EdNet: https://www.kaggle.com/competitions/riiid-test-answer-prediction/data?select=train.csv, accessed on 1 June 2024, reference [38]; Junyi Academy: https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198, accessed on 1 June 2024, reference [39]; Eedi: https://eedi.com/projects/neurips-education-challenge, accessed on 1 June 2024, reference [40].

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

Abbreviations

The following abbreviations are used in this manuscript:

A09	ASSISTment2009
A12	ASSISTment2012
ACC	Accuracy
AUC	Area Under the Curve
DKT	Deep Knowledge Tracing
DKVMN	Dynamic Key–Value Memory Networks
DLKT	Deep Learning-based Knowledge Tracing
EN	EdNet
ID	Identification
KT	Knowledge Tracing
SAINT	Separated Self-Attentive Neural Knowledge Tracing
SAKT	Self-Attentive Knowledge Tracing
SHAP	Shapley Additive Explanations
SIKT	Side Information for Knowledge Tracing
SKQ	Students, Knowledge, and Questions

References

1. Abdelrahman, G.; Wang, Q.; Nunes, B. Knowledge Tracing: A Survey. *ACM Comput. Surv.* **2023**, *55*, 1–37. [CrossRef]
2. Zanellati, A.; Mitri, D.D.; Gabbrielli, M.; Levrini, O. Hybrid Models for Knowledge Tracing: A Systematic Literature Review. *IEEE Trans. Learn. Technol.* **2024**, *17*, 1021–1036. [CrossRef]
3. Lu, Y.; Wang, D.; Chen, P.; Meng, Q.; Yu, S. Interpreting Deep Learning Models for Knowledge Tracing. *Int. J. Artif. Intell. Educ.* **2022**, *33*, 519–542. [CrossRef]
4. Gervet, T.; Koedinger, K.; Schneider, J.; Mitchell, T. When is deep learning the best approach to knowledge tracing? *J. Educ. Data Min.* **2020**, *12*, 31–54.
5. Yu, M.; Li, F.; Liu, H.; Zhang, T.; Yu, G. ContextKT: A context-based method for knowledge tracing. *Appl. Sci.* **2022**, *12*, 8822. [CrossRef]
6. Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L.J.; Sohl-Dickstein, J. Deep knowledge tracing. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9.
7. Xiong, X.; Zhao, S.; Van Inwegen, E.G.; Beck, J.E. Going deeper with deep knowledge tracing. In Proceedings of the 9th International Conference on Educational Data Mining, Raleigh, NC, USA, 2 June–2 July 2016.
8. 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, Perth, Australia, 3–7 April 2017; pp. 765–774. [CrossRef]
9. Wang, D.; Lu, Y.; Zhang, Z.; Chen, P. A generic interpreting method for knowledge tracing models. In Proceedings of the International Conference on Artificial Intelligence in Education, Durham, UK, 27–31 July 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 573–580.
10. Yang, H.; Cheung, L.P. Implicit heterogeneous features embedding in deep knowledge tracing. *Cogn. Comput.* **2018**, *10*, 3–14. [CrossRef]
11. Zhang, L.; Xiong, X.; Zhao, S.; Botelho, A.; Heffernan, N.T. Incorporating Rich Features into Deep Knowledge Tracing. In Proceedings of the Fourth (2017) ACM Conference on Learning @ Scale, Cambridge, MA, USA, 20–21 April 2017; pp. 169–172. [CrossRef]
12. Pu, S.; Becker, L. Self-Attention in Knowledge Tracing: Why It Works. In *Artificial Intelligence in Education*; Rodrigo, M.M., Matsuda, N., Cristea, A.I., Dimitrova, V., Eds.; Springer: Cham, Switzerland, 2022; pp. 731–736. [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. Suresh, S.; Ramasamy, S.; Suganthan, P.N.; Wong, C.S.Y. Incremental Knowledge Tracing from Multiple Schools. *arXiv* **2022**, arXiv:2201.06941. [CrossRef]
15. Sorrentino, P.; Petkoski, S.; Sparaco, M.; Lopez, E.T.; Signoriello, E.; Baselice, F.; Bonavita, S.; Pirozzi, M.A.; Quarantelli, M.; Sorrentino, G.; et al. Whole-brain propagation delays in multiple sclerosis, a combined tractography-magnetoencephalography study. *J. Neurosci.* **2022**, *42*, 8807–8816. [CrossRef]
16. Garcia-Bordils, S.; Mafla, A.; Biten, A.F.; Nuriel, O.; Aberdam, A.; Mazor, S.; Litman, R.; Karatzas, D. Out-of-Vocabulary Challenge Report. In Proceedings of the Computer Vision—ECCV 2022 Workshops, Tel Aviv, Israel, 23–27 October 2022; pp. 359–375. [CrossRef]

17. Chen, Y.; Wang, S.; Jiang, F.; Tu, Y.; Huang, Q. DCKT: A novel dual-centric learning model for knowledge tracing. *Sustainability* **2022**, *14*, 16307. [CrossRef]
18. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
19. AlKhuzaey, S.; Grasso, F.; Payne, T.R.; Tamma, V. Text-based question difficulty prediction: A systematic review of automatic approaches. *Int. J. Artif. Intell. Educ.* **2023**, 1–53. [CrossRef]
20. Kim, J.; Koo, S.; Lim, H. A Multi-Faceted Exploration Incorporating Question Difficulty in Knowledge Tracing for English Proficiency Assessment. *Electronics* **2023**, *12*, 4171. [CrossRef]
21. Corbett, A.T.; Anderson, J.R. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Model.-User-Adapt. Interact.* **1994**, *4*, 253–278. [CrossRef]
22. Cai, L.; Choi, K.; Hansen, M.; Harrell, L. Item response theory. *Annu. Rev. Stat. Its Appl.* **2016**, *3*, 297–321. [CrossRef]
23. Gong, Y.; Beck, J.E.; Heffernan, N.T. How to construct more accurate student models: Comparing and optimizing knowledge tracing and performance factor analysis. *Int. J. Artif. Intell. Educ.* **2011**, *21*, 27–46.
24. Pandey, S.; Karypis, G. A self-attentive model for knowledge tracing. In Proceedings of the EDM 2019—Proceedings of the 12th International Conference on Educational Data Mining, 2–5 July 2019, Montreal, QC, Canada; pp. 384–389. [CrossRef]
25. Choi, Y.; Lee, Y.; Cho, J.; Baek, J.; Kim, B.; Cha, Y.; Shin, D.; Bae, C.; Heo, J. Towards an Appropriate Query, Key, and Value Computation for Knowledge Tracing. In Proceedings of the Seventh ACM Conference on Learning @ Scale, New York, NY, USA, 12–14 August 2020; L@S '20; pp. 341–344. [CrossRef]
26. 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]
27. Dai, H.; Yun, Y.; Zhang, Y.; Zhang, W.; Shang, X. Contrastive Deep Knowledge Tracing. In Proceedings of the Artificial Intelligence in Education. Posters and Late Breaking Results, Workshops and Tutorials, Industry and Innovation Tracks, Practitioners' and Doctoral Consortium, Durham, UK, 27–31 July 2022; pp. 289–292. [CrossRef]
28. Volkovs, M.; Yu, G.; Poutanen, T. DropoutNet: Addressing Cold Start in Recommender Systems. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, California, USA, 4–9 December 2017; pp. 4964–4973.
29. Wang, Z.; Feng, X.; Tang, J.; Huang, G.Y.; Liu, Z. Deep Knowledge Tracing with Side Information. In *Artificial Intelligence in Education*; Springer: Cham, Switzerland, 2019; pp. 303–308. [CrossRef]
30. Liu, Q.; Huang, Z.; Yin, Y.; Chen, E.; Xiong, H.; Su, Y.; Hu, G. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 100–115. [CrossRef]
31. Nagatani, K.; Zhang, Q.; Sato, M.; Chen, Y.Y.; Chen, F.; Ohkuma, T. Augmenting Knowledge Tracing by Considering Forgetting Behavior. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3101–3107. [CrossRef]
32. Zhou, Y.; Li, X.; Cao, Y.; Zhao, X.; Ye, Q.; Lv, J. LANA: Towards personalized deep knowledge tracing through distinguishable interactive sequences. *arXiv* **2021**, arXiv:2105.06266. [CrossRef]
33. Pandey, S.; Srivastava, J. RKT: Relation-Aware Self-Attention for Knowledge Tracing. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 1205–1214. [CrossRef]
34. Li, L.; Wang, Z. Calibrated q-matrix-enhanced deep knowledge tracing with relational attention mechanism. *Appl. Sci.* **2023**, *13*, 2541. [CrossRef]
35. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 4–9 December 2017, Long Beach, CA, USA; pp. 6000–6010. [CrossRef]
36. Heffernan, N. Non-Skill-Builder-Data-New.csv. 2015. Available online: https://drive.google.com/file/d/0B2X0QD6q79ZJbFI2ZlRBbTk1MjQ/view?resourcekey=0-B0T0KRHYM_s7E34ur3rHoQ (accessed on 1 June 2024).
37. Heffernan, Neil. 2012-2013-Data-with-Predictions-4-Final.zip. 2020. Available online: https://drive.google.com/file/d/1cU6Ft4R3hLqA7G1rIGArVfelSZvc6RxY/view (accessed on 1 June 2024).
38. Labs, R. Riiid Answer Correctness Prediction Train.csv. 2020. Available online: https://www.kaggle.com/competitions/riiid-test-answer-prediction/data?select=train.csv (accessed on 1 June 2024).
39. Chang, H.S.; Hsu, H.J.; Chen, K.T. Modeling Exercise Relationships in E-Learning: A Unified Approach. In Proceedings of the 8th International Conference on Educational Data Mining, Madrid, Spain, 26–29 July 2015; pp. 532–535.
40. Wang, Z.; Lamb, A.; Saveliev, E.; Cameron, P.; Zaykov, Y.; Hernández-Lobato, J.M.; Turner, R.E.; Baraniuk, R.G.; Barton, C.; Jones, S.P.; et al. Diagnostic questions: The neurips 2020 education challenge. *arXiv* **2020**, arXiv:2007.12061.
41. Ni, Q.; Wei, T.; Zhao, J.; He, L.; Zheng, C. HHSKT: A learner–question interactions based heterogeneous graph neural network model for knowledge tracing. *Expert Syst. Appl.* **2023**, *215*, 119334. [CrossRef]
42. Feng, M.; Heffernan, N.; Koedinger, K. Addressing the assessment challenge with an online system that tutors as it assesses. *User Model. -User-Adapt. Interact.* **2009**, *19*, 243–266. [CrossRef]
43. Pardos, Z.A.; Baker, R.S.; San Pedro, M.; Gowda, S.M.; Gowda, S.M. Affective States and State Tests: Investigating How Affect and Engagement during the School Year Predict End-of-Year Learning Outcomes. *J. Learn. Anal.* **2014**, *1*, 107–128. [CrossRef]

44. Choi, Y.; Lee, Y.; Shin, D.; Cho, J.; Park, S.; Lee, S.; Baek, J.; Bae, C.; Kim, B.; Heo, J. Ednet: A large-scale hierarchical dataset in education. In Proceedings of the Artificial Intelligence in Education: 21st International Conference, AIED 2020, Ifrane, Morocco, 6–10 July 2020; Proceedings, Part II 21; Springer: Berlin/Heidelberg, Germany, 2020; pp. 69–73.
45. Liu, T.; Chen, W.; Chang, L.; Gu, T. Research Advances in the Knowledge Tracing Based on Deep Learning. *J. Comput. Res. Dev.* **2022**, *59*, 81–104.
46. Nohara, Y.; Matsumoto, K.; Soejima, H.; Nakashima, N. Explanation of Machine Learning Models Using Improved Shapley Additive Explanation. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, New York, NY, USA, 7–10 September 2019; BCB '19; p. 546. [CrossRef]
47. Chen, J.; Shen, J.; Long, T.; Shen, L.; Zhang, W.; Yu, Y. Heterogeneous Graph Representation for Knowledge Tracing. In Proceedings of the International Conference on Neural Information Processing, IIT Indore, India, 22–26 November 2022; pp. 224–235. [CrossRef]
48. Tan, W.; Jin, Y.; Liu, M.; Zhang, H. BiDKT: Deep Knowledge Tracing with BERT. In *Ad Hoc Networks and Tools for IT*; International Conference on Ad Hoc Networks; International Conference on Testbeds and Research Infrastructures; Springer: Cham, Switzerland, 2022; pp. 260–278. [CrossRef]
49. Huang, C.; Wei, H.; Huang, Q.; Jiang, F.; Han, Z.; Huang, X. Learning consistent representations with temporal and causal enhancement for knowledge tracing. *Expert Syst. Appl.* **2024**, *245*, 123128. [CrossRef]