TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 35/38 pp1894–1917
DOI: 10.26599/TST.2024.9010201
Volume 31, Number 3, June 2026

Research Article

LTKT: Knowledge Tracing Based on Positive and Negative Learning Transfers

Jia Xu, Rongrong Tang, Pin Lv*, Minghe Yu, Ge Yu, and Enhong Chen

Abstract: As an essential part of educational psychology, the propagated influence among pedagogical concepts (i.e., learning transfer) is important for optimizing Knowledge Tracing (KT) tasks. However, existing KT methods only consider the positive learning transfer and disregard the negative learning transfer. Thus, this paper proposes an innovative positive and negative Learning Transfer-based Knowledge Tracing model (LTKT), which makes the first attempt to concurrently utilize the positive and negative learning transfer relations among concepts to improve KT results. First, LTKT constructs a learning transfer graph. Then, a direct learning effect component and a learning transfer effect component are carefully designed in LTKT. The first component quantifies the impact of an exercise's practice result on the concept examined in the exercise. The second component, in contrast, computes the impact of the result on the concept's neighbouring concepts in the constructed learning transfer graph, considering the positive and negative learning transfer phenomena. Extensive experiments on publicity datasets demonstrate that LTKT outperforms all state-of-the-art KT methods.

Key words: Knowledge Tracing (KT); positive learning transfer; negative learning transfer; learning transfer graph; deep neural networks

1 Introduction

During the past few years, the COVID-19 closures have further expedited the adoption of Artificial Intelligence (AI) techniques to improve educational practices in Intelligent Tutoring Systems (ITS)[1]. As an example, mainstream Massive Open Online Courses (MOOC) platforms, such as edX.org, Coursera.org, and aixuexi.com, have leveraged AI technologies to predict students' learning states and provide them with personalized learning materials, which successfully improves students' learning ability[2]. Therefore,

• Jia Xu is with Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510555, China, and Key Laboratory of Cyberspace Security Defense, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China. E-mail: xujia@gzhu.edu.cn.
• Rongrong Tang is with School of Computer, Electronics and Information, Guangxi University, Nanning 530004, China. E-mail: tangrongrong@st.gxu.edu.cn.
• Pin Lv is with Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510555, China. E-mail: lvpin@gzhu.edu.cn.
• Minghe Yu is with Software College, Northeastern University, Shenyang 110819, China. E-mail: yuminghe@mail.neu.edu.cn.
• Ge Yu is with School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China. E-mail: yuge@mail.neu.edu.cn.
• Enhong Chen is with School of Computer Science, University of Science and Technology of China, Hefei 230088, China. E-mail: cheneh@ustc.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-07-10; revised: 2024-09-16; accepted: 2024-10-20

numerous AI based Knowledge Tracing (KT) models, which can predict and trace knowledge states of students using their learning logs, have been proposed recently in top conferences or journals[3–6], to enhance the personalized functionalities on MOOC platforms. Generally, a KT task consists of two main parts: (1) estimating students' knowledge states through their question-answering logs; and (2) predicting the probability of students correctly answer a new exercise at the next time step[7].

There are primarily two major categories of existing KT models: traditional models and models based on deep learning techniques. Traditional models include Bayesian Knowledge Tracing (BKT)[8], Learning Factor Analysis (LFA)[9], Performance Factor Analysis (PFA)[10], and Knowledge Tracing Machines (KTM)[11]. In the past few years, numerous efforts have been devoted to design KT models based on deep learning techniques owing to the capabilities of deep learning and have achieved better results on student performance prediction than several traditional methods[12–14]. Deep Knowledge Tracing (DKT)[12], Dynamic Key-Value Memory Networks (DKVMN)[13], and Convolutional Knowledge Tracing (CKT)[14] are representative deep knowledge tracing models proposed in the early stage. These models, however, fail to consider the learning transfer between two concepts, and thus can only generate less accurate prediction results. Recently, some researchers have perceived the importance of learning transfer for improving the effectiveness of KT[15, 16]. For example, in Ref. [15], a Graph Neural Network (GNN) based knowledge tracing model, named Graph-based Knowledge Tracing (GKT), is presented, which models the learning transfer process based on a homogeneous graph. Similarly, in Ref. [16], a Structure-based Knowledge Tracing (SKT) model is also proposed based on GNN. Unlike GKT, SKT defines two distinct types of relations among knowledge concepts, namely prerequisite and similarity relation and then builds a heterogeneous graph to represent the knowledge structure. By leveraging the diverse range of relations within the heterogeneous graph, SKT can model the learning transfer process among knowledge concepts. Although significant improvement to KT has been achieved by considering the impact of learning transfer, existing studies consider only the positive learning transfer and disregard the vital impact of negative learning transfer on KT.

As an essential part of educational psychology, learning transfer theory holds significant influence in both the educative process and learning process[17, 18], and it can be categorized as positive transfer and negative transfer[19]. Specifically, positive transfer means that learning in one context improves performance in some other context[20]. For example, it is easy to understand that the learning of addition $2+2+2=6$ facilitates the learning of multiplication $2\times 3=2+2+2=6$. Negative transfer indicates that learning in one context impacts negatively on performance in another[20]. It is known that negative transfers usually present in mathematical concepts[21]. For example, if a student learns the distributive law of multiplication (e.g., $m\,(a+b)=ma+mb$), he/she may mistakenly get $\lg\,(a+b)=\lg a+\lg b$. In conclusion, the learning transfer is a very important factor that should be considered when estimating a student's knowledge state towards a concept. On one hand, integrating the learning transfer into the knowledge state prediction can diminish the uncertainty in the knowledge state space of students, enabling teachers to devise more effective teaching strategies. On the other hand, the learning transfer is an objective phenomenon in the learning process of students, which indicates taking the learning transfer into account can enhance the accuracy of knowledge state prediction for students.

Figure 1 shows how the positive and negative transfers affect students' knowledge state. As depicted in the bottom part of the figure, six pedagogical concepts associated with a mathematics course (i.e., A, B, ..., F) are modeled by a learning transfer graph, where nodes denote pedagogical concepts and edges represent learning transfer relations among concepts. Specifically, the learning transfer relations includes positive transfer relations and negative transfer relations, which are depicted as green and red edges, respectively, in Fig. 1. Then, following the setting in Ref. [16], a positive (or negative) transfer relation can be further identified as a prerequisite relation※ (illustrated as a direct edge) or a similarity relation‡ (illustrated as an undirected edge). The middle part of Fig. 1 illustrates the learning process of a student s who practices a group of exercises sequentially, where each exercise is related to a pedagogical concept. The

※When one concept is linked to another through a prerequisite relation, it signifies that the former is regarded as the foundation for the latter.

‡Two concepts connected by a similarity relation are similar with respect to their contents.

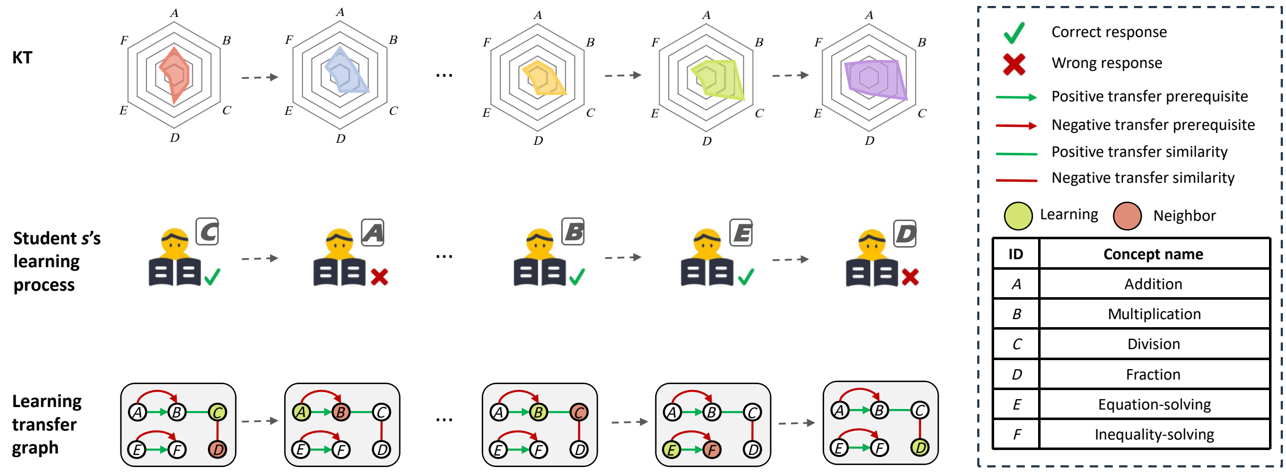


Fig. 1 Visualization showcasing the KT process for student *s*. The middle of the figure shows an exercise sequence $C \to A \to \cdots \to B \to E \to D$, where each exercise is associated with a specific concept. The radar graphs displayed at the top shows the evolving knowledge state on each concept throughout the learning process. The learning transfer graph depicted at the bottom contains four categories of learning transfer relations, which are defined in the legend. Current learning concept is highlighted in green and its neighbor concept is highlighted in red.

figure also marks the correctness of student s's answer. As depicted in the figure, after s completing the first exercise related to concept C, his/her knowledge state on concept C increases, and the updated knowledge state values for all concepts are displayed in the form of a radar graph at the top of Fig. 1. Note that when s's knowledge state of a specific concept is updated, s's knowledge state of all neighbor concepts of the concept in the learning transfer graph is updated accordingly. For example, when s's knowledge state on concept C increases, his/her knowledge state on C's neighbor concept (i.e., D) is updated to a smaller value, owing to the negative transfer relation existing between the two concepts. As another example, when s's knowledge state on concept B grows, it will trigger the augment of s's knowledge state on B's neighbor concept C, because a positive transfer relation links these two concepts. When both positive and negative transfer relations between two concepts (e.g., E and F), the adjustment of s's knowledge state to a concept (e.g., F) depends on s's knowledge state values towards all concepts. To sum up, to ensure the effectiveness of KT based on knowledge structure, it is important to consider the positive and negative learning transfer impact simultaneously on the knowledge state value of each concept in the learning transfer graph.

However, there are mainly two challenges that need to be solved.

- **Challenge 1:** given a course, how to construct a learning transfer graph for the course where pedagogical concepts in the course are linked by positive and negative transfer relations.
- **Challenge 2:** how to reasonably update the knowledge state of each concept for a student based on the knowledge structure modeled by the constructed learning transfer graph.

To solve the two challenges mentioned above, this study proposes an innovative deep learning-based KT model called positive and negative Learning Transfer-based Knowledge Tracing (LTKT), which simultaneously utilizes positive and negative learning transfer relations among pedagogical concepts to optimize the performance of KT. In order to address the first challenge, LTKT proposes a statistical approach that constructs a learning transfer graph containing multiple types of learning transfer relations for a course based on the historical interactions of enrolled students. The second challenge is dealt with careful designing of two components in LTKT, the Direct Learning Effect (DLE) component and Learning Transfer Effect (LTE) component, to ensure the reasonableness of updating a student's knowledge state on each concept based on the knowledge structure modeled by the learning transfer graph. Specifically, when a student practices an exercise, DLE component quantifies the impact of the practice result on the knowledge state of the pedagogical concept investigated by the exercise using a gated function. Subsequently, the proposed LTE component considers the impact of the student's practice result on his/her knowledge state of other neighbor concepts which are caused by positive and negative learning transfer effects. The positive and negative transfer effects are modeled using the validated viewpoint of influence propagation[16]. To optimize the impact of positive and negative transfer effects on a student's knowledge state of a concept, LTE component first employs a two-phase integration strategy to aggregate these impacts, where the sensitivity value of each student to different learning transfer effects is considered. Finally, the aforementioned gated function is utilized in the LTE component to update the knowledge state values of all neighbor concepts in the learning transfer graph based on the aggregation of the positive and negative transfer effects. This study makes several key contributions, which can be summarized as follows:

- According to current knowledge, LTKT makes the first attempt to utilise the negative transfer theory from the educational psychology domain to optimize KT.
- This study proposes a novel deep learning-based model called LTKT to achieve KT by simultaneously considering the impacts of both positive and negative transfers on students' knowledge state (or proficiency) towards every pedagogical concept. Specifically, a learning transfer graph is first constructed, based on which two components, i.e., DLE component and LTE component, are carefully designed to reasonably update the knowledge states of each student.
- Extensive experiments are conducted using real-world datasets to compare the proposed LTKT with state-of-the-art KT methods. The experimental results demonstrate the superiority and interpretability of the proposed LTKT in this paper.

The subsequent sections of this paper are structured as follows. Related works are analyzed in Section 2. Section 3 discusses the preliminary of this study. The proposed LTKT model is elaborated upon in Section 4. The experimental methodology and results are presented in Section 5. Finally, Section 6 presents the conclusions, limitations and future work for continuing this research.

2 Related Work

In this section, we conduct an analysis of related works from the following three aspects.

2.1 KT without considering knowledge structure

KT is employed to evaluate students' knowledge states based on their historical interactions. Its primary usage is to predict whether a student will provide a correct response to the next exercise[22]. Most existing KT works can be categorized into either traditional KT methods or deep learning-based KT methods. BKT is the earliest proposed KT model[8]. It is a probabilistic model which utilizes a hidden Markov model to trace a student's evolving knowledge state as a set of binary variables (i.e., know or do not know). Some traditional models are proposed based on logistic functions[9−11]. These models use a logistic function to transform the degree of skill mastery into the estimate of the probability of correct answer[23]. DKT is the first KT model based on deep learning[12]. Specifically, DKT applies a Recurrent Neural Network (RNN) to achieve effective KT with respect to the whole knowledge space, but cannot provide the mastery degree information targeting a concept. Yeung and Yeung[24] further improved DKT by addressing two problems, i.e., the reconstruction problem and the wavy transition problem in prediction that exist in DKT. Gating-controlled Forgetting and Learning mechanisms for Deep Knowledge Tracing (GFLDKT)[25] is built based on the design pattern of LSTM and it designs two gating-controlled mechanisms to model the forgetting and learning behaviors in students' learning process. Liu et al.[26] improved the prediction performance of DKT with two auxiliary learning tasks, i.e., question tagging prediction task and individualized prior knowledge prediction task. Multiple Question Factors for Knowledge Tracing (MQFKT)[27] improves DKT by the following two aspects. On one hand, a calibrated student-concept connection space is designed to achieve more fine-grained learning interaction representation. On the other hand, exercise difficulty level and discrimination are introduced to obtain more accurate prediction results. To overcome the shortcoming of DKT, the DKVMN[13] is presented, which applies Memory-Augmented Neural Networks (MANN) to quantify the knowledge state for each latent concept during the KT procedure. Specifically, DKVMN creates a static matrix called key matrix to store latent knowledge concepts and a dynamic matrix called value matrix to store and update the mastery degree values of those knowledge concepts. Zhao and Sahebi[28] enhanced the DKVMN model and proposed the Graph-enhanced Multi-activity Knowledge Tracing (GMKT) model, which utilizes the information of both assessed activities (i.e., practice exercises) and non-assessed activities (i.e., watching video lectures) to guide prediction of the knowledge state in KT. Kou et al.[29] introduced a learner's carelessness rate into DKVMN, successfully improving the prediction accuracy of it. Tsutsumi et al.[30] added an exercise network to DKVMN so as to calculate the difficulty of each exercise, achieving the goal of independent learning of student parameters and exercise parameters. Meanwhile, a hypernetwork was introduced by them to improve their proposal, ultimately enhancing the prediction accuracy of DKVMN. Zhang et al.[31] optimized DKVMN by considering students' knowledge-absorption ability and problem-solving ability. CKT[14] is another popular deep learning-based KT method that first employs Convolutional Neural Networks (CNNs) to handle the KT problem[2]. CKT measures the individualized prior knowledge of student using his/her historical interactions and designs hierarchical convolutional layers to extract their individualized learning rates based on their continuous learning interactions. Although each of the deep learning based KT methods, namely DKT, DKVMN, and CKT, achieves a significant performance improvement compared to the traditional KT methods. They assume that the knowledge concepts are independent of each other or simply model the exercise-concept interaction information, whereas latest works argue that the knowledge structure is useful for ensuring the effectiveness of KT.

2.2 KT based on knowledge structure

It is well recognized that knowledge concepts are naturally linked to each other, forming a knowledge structure. The knowledge structure is generally modeled by a graph. In the graph, vertexes symbolize the concepts, while edges depict the relations among the concepts[32]. Recently, a growing number of studies have proposed to use knowledge structure information to optimize KT. For example, Gan et al.[32] augmented exercise embedding based on a knowledge structure to ensure that the learned exercise representation vectors carry concept-concept relations. Chen et al.[33] and Wang et al.[34] further refined concept-concept relations into prerequisite relations and similarity relations, base on which a regularization term is proposed to better predict students' knowledge states. Existing KT models that employ knowledge structure have shown promising results for KT. However, they ignore learning transfer among concepts, which is a common phenomenon in the learning process[35]. Under such condition, if a student has never practiced exercises investigating a certain concept, these models will simply deem the student's knowledge state with respect to the concept as 0, which does not accurately reflect the actual circumstances.

2.3 KT based on learning transfer

Learning transfer is widely presented in the learning process of knowledge and skills. According to educational psychology theories, learning transfer refers to the influence of learning one concept on learning another concept[17]. Recent research works, such as GKT[15], SKT[16], Knowledge Tracing model with Learning Transfer (KTLT)[36], Context-based Knowledge Tracing (ContextKT)[37], Knowledge Structure enhanced Graph Knowledge Tracing (KSGKT)[32], CAKT[38], and sparseKT[39] target the learning transfer effect between pedagogical concepts when implementing KT task. The KT task is reformulated in GKT as an application of GNN and assumes that concept-concept relations are modeled by a homogeneous graph. GKT then models the learning transfer effect among concepts based on a homogeneous graph. Similar to GKT, KTLT also models the concept-concept relations as a homogeneous graph and utilizes the positive learning transfer relations between two concepts to optimize its KT tasks. However, KTLT designs under the assumption that a student's knowledge state remains stable within a time window t, limiting its ability to trace real-time changes in a student's knowledge state. Different from GKT, SKT assumes two types of concept-concept relations, namely prerequisite and similarity, and utilizes a heterogeneous graph structure to model such abundant types of relations, based on which the learning transfer effect among concepts are considered. Furthermore, many studies have focused on positive transfer between historical concepts and current concept[32, 37–39]. These studies consistently assume that a student's knowledge state on a concept investigated by the current exercise is related to their knowledge states on similar concepts which are investigated by historical exercises. They implement this idea by employing the attention networks which uses the softmax function to scale the impact of each historical concepts on current concept. Nevertheless, all the abovementioned studies only consider the effect of positive learning transfer among pedagogical concepts and ignore the effect of negative learning transfer, which limits their performance to certain extent. These efforts are based on the idealized assumption that students can exclude the effect of negative transfer and make complete use of positive transfer to facilitate their learning. However, in the actual learning process, the influence of negative transfer is inevitable.

The research gaps between this study and the related works are outlined in Table 1.

3 Preliminary

In this section, we begin by providing an elaborate description of a learning transfer graph and then provide its precise definition. Subsequently, we formally define the KT problem based on learning transfer graph. The notations employed throughout this study are explicated and summarized in Table 2.

3.1 Learning transfer graph

Based on learning transfer theory, relations between pedagogical concepts can be classified as positive transfer relations and negative transfer relations. Recent studies[16, 32–34] indicate that a positive (or negative) transfer relation can be further identified as a prerequisite or similarity relation. In this study, we define a Learning Transfer Graph (LTG (VX, EG)) considering these four types of transfer relations. An example of an LTG (VX, EG) is shown at the bottom of Fig. 1. A detailed explanation of each type of transfer relation and its representation in the LTG (VX, EG) is provided below.

Table 1 Research gap between this study and related works (×: aspect excluded; √: aspect included).

Method	Knowledge structure	Positive transfer	Negative transfer
DKT[12]	×	×	×
DKVMN[13]	×	×	×
CKT[14]	×	×	×
GKT[15]	√	√	×
SKT[16]	√	√	×
DKT+ [24]	×	×	×
GFLDKT[25]	×	×	×
AT-DKT[26]	×	×	×
MQFKT[27]	×	×	×
GMKT[28]	×	×	×
DKVMN-C[29]	×	×	×
Proposed-HN[30]	×	×	×
DKVMN-KAPS[31]	×	×	×
KSGKT[32]	√	√	×
PDKT[33]	√	×	×
DKTS[34]	√	×	×
KTLT[36]	√	√	×
ContextKT[37]	√	√	×
CAKT[38]	√	√	×
SPARSEKT[39]	√	√	×
LTKT (ours)	√	√	√

- **Positive Transfer Prerequisite Relation (PTPR).** It is illustrated by a green directed edge in the LTG (VX, EG), where the former concept in the relation is considered the foundation of the latter concept (i.e., the concept being directed) and mastering the former concept has a promoting effect on the study of the latter concept.
- **Positive Transfer Similarity Relation (PTSR).** It is represented as a green undirected edge in the LTG (VX, EG), where the two concepts linked by the edge are similar in content and the learning of one concept promotes the learning of the other concept.
- **Negative Transfer Prerequisite Relation (NTPR).** It is denoted by a red directed edge in the LTG (VX, EG). Such a relation can be interpreted as the former concept in the relation as the basis of the latter concept, however, the study of the former interferes with the study of the latter.
- **Negative Transfer Similarity Relation (NTSR).** It is denoted as a red undirected edge in the LTG (VX, EG). The NTSR relation indicates that though two concepts linked together are similar, the learning of one concept inhibits the learning of the other.

Definition 1 (Learning transfer graph) A learning transfer graph defined over a set of N concepts $C=\{c_1, c_2, \ldots, c_n, \ldots, c_N\}$ is formalized as a graph LTG (VX, EG) with an edge type mapping function $\psi: \mathrm{EG} \rightarrow$ {PTPR, PTSR, NTPR, NTSR}. Here, $\mathrm{VX}=\{\mathrm{vx}_1, \mathrm{vx}_2, \ldots, \mathrm{vx}_n, \ldots, \mathrm{vx}_N\}$ represents N vertexes in the graph with every vertex $\mathrm{vx}_n \in \mathrm{VX}$ corresponding to one concept $c_j \in C$. $\mathrm{EG}=\{\mathrm{eg}_1, \mathrm{eg}_2, \ldots, \mathrm{eg}_m, \ldots, \mathrm{eg}_M\}$ represents the set of edges within the graph, with each edge having a type from {PTPR, PTSR, NTPR, NTSR}.

3.2 KT Problem with learning transfer graph

The objective of KT is to trace changes in a student's knowledge state towards pedagogical concepts based on the student's historical interactions. Thus, KT can be used to predict student performances in future exercises. Following the settings in Refs. [6, 12, 14, 16,

Table 2 Frequently-used notations and their descriptions.

Notation	Description
LTG (VX, EG)	Learning transfer graph with a vertex set VX and an edge set EG
$\mathrm{VX} = \{\mathrm{vx}_1, \mathrm{vx}_2, \ldots, \mathrm{vx}_n, \ldots, \mathrm{vx}_N\}$	Vertex set has N vertexes and vx_n denotes the n-th vertex in the set
$\mathrm{EG} = \{\mathrm{eg}_1, \mathrm{eg}_2, \ldots, \mathrm{eg}_m, \ldots, \mathrm{eg}_M\}$	Edge set owns M edges and eg_m is the m-th edge in the set
$S = \{s_1, s_2, \ldots\ s_i, \ldots\ s_I\}$	Student set has I students and s_i is the i-th student in the set
$\mathrm{EX} = \{\mathrm{ex}_1, \mathrm{ex}_2, \ldots\ \mathrm{ex}_j, \ldots\ \mathrm{ex}_J\}$	Exercise set has J exercises and ex_j denotes the j-th exercise in the set
$C = \{c_1, c_2, \ldots, c_n, \ldots, c_N\}$	Concept set has N pedagogical concepts and c_n is the n-th concept in the set
$X = \{x_1, x_2, \ldots, x_t, \ldots, x_T\}$	Student's learning sequence owns T learning interactions. The t-th interaction $x_t = (\mathrm{ex}_t, a_t)$ records the student's correctness $a_t \in \{0, 1\}$ when answers the exercise ex_t
$\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t, \ldots, \boldsymbol{y}_T\}$	Sequence records a student's T knowledge states with respect to T time steps
$\boldsymbol{y}_t = \left\{\tilde{\boldsymbol{h}}^t_{c_1}, \tilde{\boldsymbol{h}}^t_{c_2}, \ldots, \tilde{\boldsymbol{h}}^t_{c_n}, \ldots, \tilde{\boldsymbol{h}}^t_{c_N}\right\}$	The t-th knowledge state in $\mathcal{Y}$, where $\boldsymbol{h}^t_{c_n}$ is the student's knowledge state for the n-th concept in C with respect to the t-th time step
$\boldsymbol{M}_x$	Embedding matrix of a learning interaction
$\boldsymbol{v}_t$	Direct learning effect vector obtained by embedding $x_t = (\mathrm{ex}_t, a_t)$
$\mathcal{G}$	Gated function
$\boldsymbol{h}^t_{c_i}$	Student's knowledge state towards the concept c_i, which is computed considering the direct learning effect
$\boldsymbol{e}\,(c_i)$	Embedding vector of concept c_i in C
$\mathbf{PTPR}_{c_i \to c_j}$	Computed positive transfer prerequisite effect brought by concept c_i to concept c_j
$\mathbf{PTSR}_{c_i \to c_j}$	Computed positive transfer similarity effect brought by concept c_i to concept c_j
$\mathbf{PTSR}_{\mathcal{N}_{\mathrm{PTSR}}(c_i) \to c_i}$	Computed positive transfer similarity effect brought by concept c_i's neighbors to concept c_i
$\mathbf{NTPR}_{c_i \to c_j}$	Vomputed negative transfer prerequisite effect brought by concept c_i to concept c_j
$\mathbf{NTSR}_{c_i \to c_j}$	Computed negative transfer similarity effect brought by concept c_i to concept c_j
$\mathbf{NTSR}_{\mathcal{N}_{\mathrm{NTSR}}(c_i) \to c_i}$	Computed negative transfer similarity effect brought by concept c_i's neighbors to concept c_i
α	Weight used to calculate the weighted sum of prerequisite effect and similar effect
$\boldsymbol{A}^+_{c_n}$	Aggregation result of all positive transfer effects
$\boldsymbol{A}^-_{c_n}$	Aggregation result of all negative transfer effects
β	Sensitivity of a student to different learning transfer effects
$\boldsymbol{A}_{c_n}$	Weighted sum of positive transfer effect and negative transfer effect
$\boldsymbol{I}^t_{c_n}$	Aggregation result of all positive transfer effects and all negative transfer effects
f_{out}	Function that maps a knowledge state to a predictive probability

40], each exercise is assumed to relate to one pedagogical concept during the KT procedure. A formal definition of the KT problem with learning transfer graph which is investigated in this study is provided below.

Definition 2 (KT problem with learning transfer graph) Let $X = \{x_1, x_2, \ldots, x_t, \ldots, x_T\}$ represent a student's historical interaction sequence, where the interaction for each time step is denoted as $x_t = (\mathrm{ex}_t, a_t)$ and $a_t \in \{1, 0\}$ indicates the binary correctness to exercise ex_t at the time step t. The objective of the KT problem with a learning transfer graph LTG (VX, EG) built over a set of N pedagogical concepts $C = \{c_1, c_2, \ldots, c_n, \ldots, c_N\}$ is to model a student's knowledge state $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_t, \ldots, \boldsymbol{y}_T\}$ for N concepts at each time step, and predict the probability that the student will correctly answer a new exercise ex_{t+1} at the time step $t+1$, i.e., $\Pr(a_{t+1} = 1 \mid \mathrm{ex}_{t+1}, X, \mathrm{LTG}\,(\mathrm{VX}, \mathrm{EG}))$.

4 LTKT Model

Within this section, we initially offer a comprehensive overview of LTKT, followed by detailed elaboration on each component implemented in LTKT.

4.1 Overview

The LTKT model leverages both positive and negative transfer relations among pedagogical concepts to improve the deep KT performance. Figure 2 depicts the architecture of the LTKT model, which contains four major components, namely Learning Transfer Graph Construction (LTGC) component, DLE component,

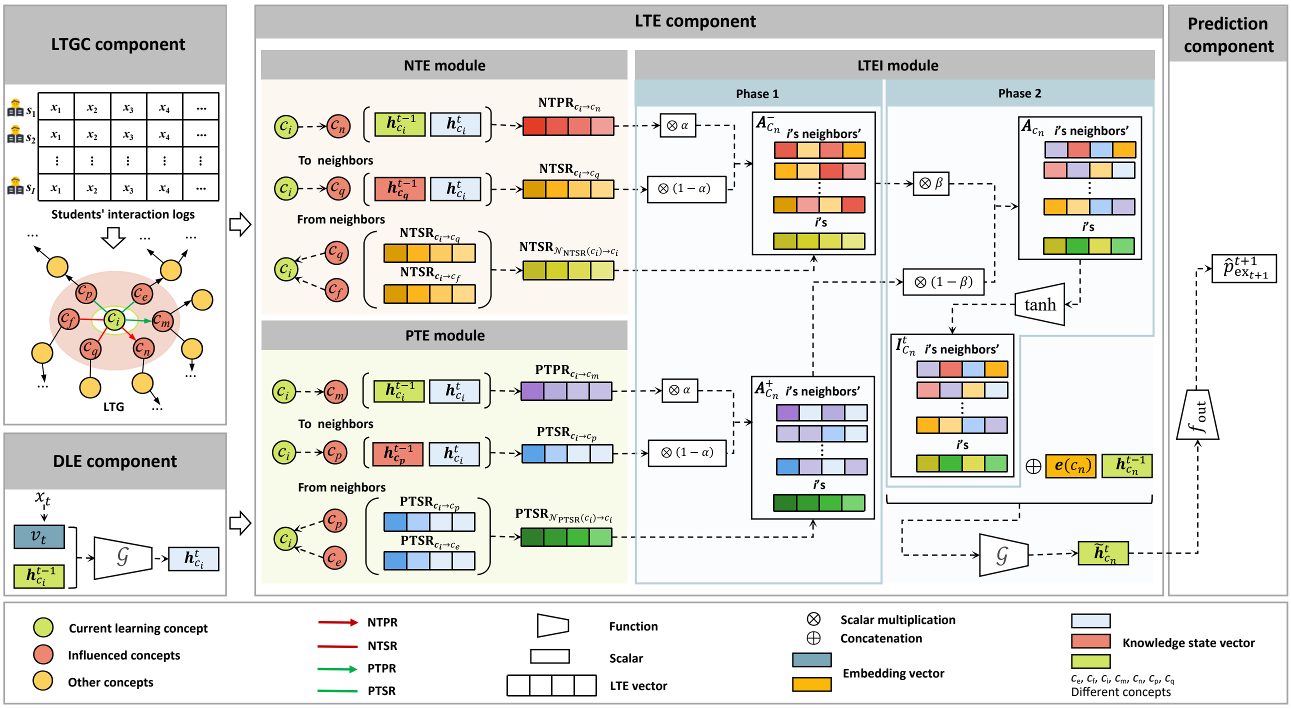


Fig. 2 Comprehensive architecture of the proposed LTKT model. The mathematical notations used in this figure are presented in Table 2.

LTE component, and prediction component. Particularly, the LTE component has three modules: Positive Transfer Effect (PTE) module, Negative Transfer Effect (NTE) module, and Learning Transfer Effect Integration (LTEI) module.

As shown in Fig. 2, at the beginning, the LTGC component in LTKT is responsible for building a learning transfer graph, which explicitly models four types of learning transfer relations among all pedagogical concepts. Subsequently, the DLE component is in charge of modeling the direct learning effect among concepts, based on which this component updates the knowledge state of a target student according to the direct learning effect. Using the learning transfer graph constructed by the LTGC component and the knowledge state of the target student updated via the DLE component as the inputs, the LTE component first models the two learning transfer effects, that is, negative transfer effect and positive transfer effect, and then uses a two-phase integration strategy to fuse these two effects. The fused result is then passed through a gated function $\mathcal{G}$ to further update the knowledge state of target student based on the two learning transfer effects. Finally, the prediction component in the LTKT model is implemented to predict whether the target student will be able to answer the subsequent exercise correctly leveraging his/her updated knowledge state.

4.2 LTGC component

The LTGC component is responsible for construct a learning transfer graph LTG (VX, EG) for a given course based on historical interactions of students involved in the course. In a constructed LTG (VX, EG), each vertex corresponds to a certain pedagogical concept of the course, and an edge models the learning transfer relation between a pair of concepts. In general, the pedagogical concepts of the course are easily accessible, while determining the learning transfer relations among concepts is a non-trivial task. Therefore, the LTGC component designs a statistical method to mine learning transfer relations among concepts by analyzing the statistics of students' historical interaction events. Notably, the recency effect is taken into account during this analysis. In particular, the recency effect indicates that the students tend to forget their previous knowledge, as proven in Refs. [4–6]. Therefore, in our setting, the student's historical interaction events that are not adjacent to the current time step are considered to have no impact on the interaction events that occur at the current time. In the following, we provide details on mining positive transfer relations and negative transfer relations among vertexes in an LTG (VX, EG). For the convenience of discussion, let "$i\checkmark$" denotes the interaction event that a student correctly answers the exercise in relation to the concept c_i, and let "$i\times$" represents the interaction event that a student gives a wrong answer to the exercise with respect to the concept c_i. Then, "$i\checkmark \rightarrow j\times$" is used to represent that the interaction event "$j\times$" occurs after the interaction event "$i\checkmark$".

According to positive transfer theory, if the learning of concept c_i enhances the learning of concept c_j, a positive transfer relation exists within the pair of concepts. We find through observation and analysis that the interaction event "$i\checkmark \rightarrow j\checkmark, (i \neq j)$" in students' historical learning logs match well with the positive transfer relation. The frequency value of each instance of such interaction event which is computed over the students' historical learning logs, is used to determine the positive transfer relations in LTG (VX, EG). To consider the recency effect, the event "$j\checkmark$" is supposed to take place immediately after the event "$i\checkmark$".

As for negative transfer theory, its connotation involves two aspects: (1) the learning of concept c_i interferes the learning of concept c_j; and (2) the decline in the proficiency of concept c_i decreases the cognitive confusion between the two concepts c_i and c_j, leading to an increase in the proficiency of concept c_j. Following such connotation, the computed frequency values of instances of both interaction events, that is, "$i\checkmark \rightarrow j\times, (i \neq j)$" and "$i\times \rightarrow j\checkmark, (i \neq j)$", are considered when determining a negative transfer relation between a pair of concepts. Similarly, the event "$j\times$" (or "$j\checkmark$") is assumed to take place immediately after the event "$i\checkmark$" (or "$i\times$") in each of the two events, to ensure the recency effect is considered.

After determining a positive (or negative) transfer relation between a pair of concepts, the idea in Ref. [16] is applied to the LTGC component to further identify whether the relation is a prerequisite or similarity relation. The specific calculation method for each type of edge in the LTG (VX, EG), that is, PTPR, PTSR, NTPR, and NTSR, are discussed in Section 5.1.2.

4.3 DLE component

Given an exercise that investigates a certain pedagogical concept, the DLE component aims to quantify the influence of a student's practice result of the exercise on the knowledge state of the concept. As shown in the bottom left part of Fig. 2, based on a learning interaction $x_t = (\text{ex}_t^{c_i}, a_t)$ which represents a student getting the practice result $a_t \in \{0,1\}$ towards the exercise ex_t investigated concept c_i at time step t, the DLE component can quantify the influence of the practice result on the knowledge state of concept c_i as a vector $\boldsymbol{v}_t$. Inspired by Refs. [12, 16, 24, 41], to compute $\boldsymbol{v}_t$, we first convert the learning interaction $x_t = (\text{ex}_t^{c_i}, a_t)$ into a one-hot vector $\tilde{\boldsymbol{x}}_t \in \{0,1\}^{2N}$, which is determined by

$$\tilde{\boldsymbol{x}}_t[j] = \begin{cases} 1, & \text{if } j = 2 \times \text{ex}_t + a_t; \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The quantified impact of the direct learning effect $\boldsymbol{v}_t$ is then computed as

$$\boldsymbol{v}_t = \tilde{\boldsymbol{x}}_t \boldsymbol{M}_x \tag{2}$$

where $\boldsymbol{M}_x \in \mathbf{R}^{2N \times d_v}$ denotes an embedding matrix, N signifies the total number of concepts, and d_v is the dimension of $\boldsymbol{v}_t$.

Let $\boldsymbol{h}_{c_i}^{t-1}$ be the student's knowledge state at previous time step $t-1$ and $\boldsymbol{h}_{c_i}^{t}$ be the student's knowledge state at time step t, which considers the direct learning effect brought by his/her learning interaction $x_t = (\text{ex}_t^{c_i}, a_t)$. Then, with $\boldsymbol{h}_{c_i}^{t-1}$ and the quantified impact of direct learning effect $\boldsymbol{v}_t$ as the inputs, a function of Gated Recurrent Unit (GRU) gate[42] $\mathcal{G}(\cdot)$, which is shown in Eq. (3), is utilized to simulate the impact of direct learning effect on the knowledge state of student with respect to concept c_i,

$$\boldsymbol{h}_{c_i}^{t} = \mathcal{G}(\boldsymbol{v}_t, \boldsymbol{h}_{c_i}^{t-1}) \tag{3}$$

4.4 LTE component

The LTE component is employed to further update the knowledge state for each concept of a student based on the learning transfer relations among pedagogical concepts modeled by the learning transfer graph LTG (VX, EG) built in the LTGC component. In contrast to the direct learning effect handled by the DLE component, which models the evolution of the knowledge state of a target concept c_i, the LTE models the evolution of the knowledge state of other concepts triggered by the change of the knowledge state of the target concept c_i. As shown in Fig. 2, the LTE component applies two modules, a PTE module and an NTE module, to ensure the reasonableness of updating a student's knowledge state on every concept based on the LTE among concepts.

4.4.1 PTE module

The PTE module, which is shown at the bottom of the LTE component in Fig. 2, quantifies the impacts of PTPR and PTSR defined in a learning transfer graph LTG (VX, EG), which are then utilized to update a student's knowledge states towards pedagogical concepts.

(1) Positive transfer prerequisite effect: The positive transfer prerequisite effect is caused by PTPR. A PTPR is represented as a green directed edge in an LTG (VX, EG) to indicate that the knowledge learned by a student with respect to the predecessor concept benefits the learning of the successor concept. Therefore, to model the positive transfer prerequisite effect, a unidirectional transfer method is used to transfer the learning performance of the predecessor concept to its successor concepts along the directed PTPR edges. Specifically, as shown in Fig. 2, after the knowledge state of concept c_i evolving from $\boldsymbol{h}_{c_i}^{t-1}$ to $\boldsymbol{h}_{c_i}^{t}$ owing to the direct learning effect, the learned knowledge of concept c_i at time step t will transfer along the directed PTPR edge to c_i's successor c_m. Given a PTPR edge $c_i \rightarrow c_j$, where c_i is the predecessor concept and c_j is c_i's successor concept, the positive transfer prerequisite effect from concept c_i to concept c_j which is denoted by $\mathbf{PTPR}_{c_i \rightarrow c_j} \in \mathbf{R}^{d_h}$ is quantified as

$$\begin{aligned}\mathbf{PTPR}_{c_i \rightarrow c_j} &= \mathrm{RELU}\left(\boldsymbol{W}_{\mathrm{pp}} \boldsymbol{Q}_{c_i \rightarrow c_j} + \boldsymbol{b}_{\mathrm{pp}}\right),\\ \boldsymbol{Q}_{c_i \rightarrow c_j} &= \boldsymbol{h}_{c_i}^{t} - \boldsymbol{h}_{c_j}^{t-1},\\ &\forall c_j \in \mathcal{S}_{\mathrm{PTPR}}(c_i)\end{aligned} \tag{4}$$

where $\mathcal{S}_{\mathrm{PTPR}}(c_i)$ is a successor function that can return all successor concepts of concept c_i in terms of PTPR, d_h is the dimensionality of the krowledge state vector h, and $\boldsymbol{W}_{\mathrm{pp}} \in \mathbf{R}^{d_h \times d_h}$ and $\boldsymbol{b}_{\mathrm{pp}} \in \mathbf{R}^{d_h}$ are the learned parameters.

(2) Positive transfer similarity effect: The existence of PTSR among vertexes in an LTG (VX, EG) leads to the positive transfer similarity effect. A PTSR is denoted as a green undirected edge in an LTG (VX, EG), which indicates that the knowledge learned in one concept promotes the learning of the other concept connected by a PTSR. Therefore, to model such positive transfer similarity effect, a bidirectional transfer method is applied to transfer the learning performance of one concept to other concepts along the undirected PTSR edges. Specifically, as shown in Fig. 2, after the knowledge state of concept c_i changes from $\boldsymbol{h}_{c_i}^{t-1}$ to $\boldsymbol{h}_{c_i}^{t}$ owing to the direct learning effect, the learned knowledge of concept c_i at time step t is firstly transferred along two undirected PTSR edges to c_i's PTSR neighbors c_p and c_e separately. Wang et al.[34] and Tong et al.[16] found that, in similar relations, the enhancement of a specific concept's proficiency will reversely promote the proficiency of its PTSR neighbor concept and vice versa, which leads to a comparable level of proficiency among the neighbor concepts. Therefore, after updating the knowledge states of concept c_i's PTSR neighbors based on the learned knowledge of concept c_i at time step t, the knowledge state of concept c_i is in turn updated by absorbing the aggregation of those quantified positive transfer similarity effects sent by c_i and received by all PTSR neighbors of concept c_i. This consequently leads to neighbor concepts having similar proficiency. To summarize, there exist two types of positive transfer similarity effect between a concept c_i and its PTSR neighbor concepts. Among them, the first type is the positive transfer similarity effect from concept c_i to concept c_j, which is denoted by: $\mathbf{PTSR}_{c_i \rightarrow c_j} \in \mathbf{R}^{d_h}$ and is quantified as

$$\begin{aligned}\mathbf{PTSR}_{c_i \rightarrow c_j} &= \mathrm{RELU}\left(\boldsymbol{W}_{\mathrm{ps}} \boldsymbol{U}_{c_i \rightarrow c_j} + \boldsymbol{b}_{\mathrm{ps}}\right),\\ \boldsymbol{U}_{c_i \rightarrow c_j} &= \boldsymbol{h}_{c_i}^{t} \oplus \boldsymbol{h}_{c_j}^{t-1},\\ &\forall c_j \in \mathcal{N}_{\mathrm{PTSR}}(c_i)\end{aligned} \tag{5}$$

where the function $\mathcal{N}_{\mathrm{PTSR}}(c_i)$ returns the set of PTSR neighbors of concept c_i, $\boldsymbol{W}_{\mathrm{ps}} \in \mathbf{R}^{d_h \times d_h}$ and $\boldsymbol{b}_{\mathrm{ps}} \in \mathbf{R}^{d_h}$ are learned parameters, and $\oplus$ is the concatenation operation.

The second type is the positive transfer similarity effect from PTSR neighbors to c_i, which is denoted by $\mathbf{PTSR}_{\mathcal{N}_{\mathrm{PTSR}}(c_i) \rightarrow c_i}$. Specifically, $\mathbf{PTSR}_{\mathcal{N}_{\mathrm{PTSR}}(c_i) \rightarrow c_i} \in \mathbf{R}^{d_h}$ and is computed as

$$\mathbf{PTSR}_{\mathcal{N}_{\mathrm{PTSR}}(c_i) \rightarrow c_i} = \sum_{c_j \in \mathcal{N}_{\mathrm{PTSR}}(c_i)} \mathbf{PTSR}_{c_i \rightarrow c_j} \tag{6}$$

where $\mathcal{N}_{\mathrm{PTSR}}(c_i)$ is the neighborhood function that returns all PTSR neighbors of concept c_i.

4.4.2 NTE module

Previous studies on KT neglect the impacts of negative transfer relations among concepts on the prediction of students' knowledge states, while the NTE among concepts widely presents in the learning process[35]. The NTE module, shown in the upper part of the LTE component in Fig. 2, quantifies the impacts of both the negative transfer prerequisite relations and negative transfer similarity relations in an LTG (VX, EG), which are subsequently used to update a student's knowledge state in relation to every pedagogical concept.

(1) Negative transfer prerequisite effect. The negative transfer prerequisite effect is caused by the NTPR in an LTG (VX, EG). An NTPR is represented as a red directed edge in LTG (VX, EG) to indicate that the knowledge learned by a student with regard to the predecessor concept will interfere or suppress the learning of the successor concept in an NTPR. Similar to the modelling of positive transfer prerequisite effect, a unidirectional transfer method is used to model the negative transfer prerequisite effect which transfers the learning performance of the predecessor concept to its successor concepts along the directed NTPR edges. Specifically, as shown in Fig. 2, after the knowledge state of concept c_i evolving from $\boldsymbol{h}_{c_i}^{t-1}$ to $\boldsymbol{h}_{c_i}^{t}$ due to the direct learning effect, the learned knowledge of concept c_i at time step t will transfer along the directed NTPR edge to c_i's successor c_n. Given an NTPR edge $c_i \rightarrow c_j$, where c_i is the predecessor concept and c_j is c_i's successor concept. The negative transfer prerequisite effect from concept c_i to concept c_j which is expressed by $\mathbf{NTPR}_{c_i \rightarrow c_j} \in \mathbf{R}^{d_h}$ is quantified as

$$\begin{aligned} &\mathbf{NTPR}_{c_i \rightarrow c_j} = \mathrm{RELU}(\boldsymbol{W}_{\mathrm{np}} \boldsymbol{R}_{c_i \rightarrow c_j} + \boldsymbol{b}_{\mathrm{np}}), \\ &\boldsymbol{R}_{c_i \rightarrow c_j} = \boldsymbol{h}_{c_i}^{t} - \boldsymbol{h}_{c_j}^{t-1}, \\ &\forall c_j \in \mathcal{S}_{\mathrm{NTPR}}(c_i) \end{aligned} \tag{7}$$

where $\mathcal{S}_{\mathrm{NTPR}}(c_i)$ is a successor function that can return all successor concepts of concept c_i in terms of NTPR. $\boldsymbol{W}_{\mathrm{np}} \in \mathbf{R}^{d_h \times d_h}$ and $\boldsymbol{b}_{\mathrm{np}}$ are learned parameters.

(2) Negative transfer similarity effect. The negative transfer similarity effect is caused by the NTSR. An NTSR is denoted as a red undirected edge in an LTG (VX, EG) to indicate that the knowledge learned by a student in a certain concept will interfere the other concept in an NTSR, and vice versa. Thereafter, similar to the modelling of positive transfer similarity effect, a unidirectional transfer method is used to model the negative transfer similarity effect. Such method targets at transferring the learning performance of a concept along the undirected NTSR edges to other concepts. Specifically, as shown in Fig. 2, after the knowledge state of concept c_i changes from $\boldsymbol{h}_{c_i}^{t-1}$ to $\boldsymbol{h}_{c_i}^{t}$ owing to the direct learning effect, the learned knowledge of concept c_i at time step t will first transfer along the undirected NTSR edge to concept c_i's NTSR neighbors c_q and c_f. Subsequently, the knowledge state of concept c_i is in turn updated by absorbing the summation of those quantified negative transfer similarity effects sent by c_i and received by all NTSR neighbors of c_i, which leads to the similar proficiency of the neighbor concepts. As a summary, there are two types of negative transfer similarity effect between concept c_i and its NTSR neighbors. Among them, the first type is the negative transfer similarity effect from c_i to c_j which is denoted by $\mathbf{NTSR}_{c_i \rightarrow c_j} \in \mathbf{R}^{d_h}$ and quantified as

$$\begin{aligned} &\mathbf{NTSR}_{c_i \rightarrow c_j} = \mathrm{RELU}\left(\boldsymbol{W}_{\mathrm{ns}} \boldsymbol{D}_{c_i \rightarrow c_j} + \boldsymbol{b}_{\mathrm{ns}}\right), \\ &\boldsymbol{D}_{c_i \rightarrow c_j} = \boldsymbol{h}_{c_i}^{t} \oplus \boldsymbol{h}_{c_j}^{t-1}, \\ &\forall c_j \in \mathcal{N}_{\mathrm{NTSR}}(c_i) \end{aligned} \tag{8}$$

where $\mathcal{N}_{\mathrm{NTSR}}(c_i)$ represents a neighborhood function that retrieves all NTSR neighboring concepts of concept c_i. $\boldsymbol{W}_{\mathrm{ns}} \in \mathbf{R}^{d_h \times d_h}$ and $\boldsymbol{b}_{\mathrm{ns}}$ are learned parameters. $\oplus$ indicates the concatenate operation.

The second type is the negative transfer similarity effect transferred from c_j to c_i which is denoted by $\mathbf{NTSR}_{\mathcal{N}_{\mathrm{NTSR}}(c_i) \rightarrow c_i} \in \mathbf{R}^{d_h}$ and quantified as

$$\mathbf{NTSR}_{\mathcal{N}_{\mathrm{NTSR}}(c_i) \rightarrow c_i} = \sum_{c_j \in \mathcal{N}_{\mathrm{NTSR}}(c_i)} \mathbf{NTSR}_{c_i \rightarrow c_j} \tag{9}$$

where $\mathcal{N}_{\mathrm{NTSR}}(c_i)$ is the same neighborhood function as Eq. (8).

In summary, in addition to considering the impact of direct learning effect on the prediction of current knowledge state for a certain concept, we also consider learning transfer in the LTE component so that the knowledge states of each student can be further updated according to both the positive transfer relations and negative transfer relations modeled by an LTG (VX, EG). Specifically, (1) we mine the learning transfer graph LTG (VX, EG) from all students' historical interaction data and model the impact of positive and negative transfer effects on students' knowledge states; and (2) given a certain concept c_i, when modeling its positive (or negative) transfer prerequisite effect, the amount of transfer effect is only determined by the change of proficiency in concept c_i. However, when modeling its positive (or negative) transfer similarity effect, the amount of transfer effect is co-determined by the knowledge state of concept c_i and its neighbor concepts in the LTG (VX, EG).

4.4.3 LTEI module

As shown in Fig. 2, to quantify the impact of positive transfer effects and negative transfer effects on a student's knowledge state in a concept, an LTEI module is proposed, which owns two processing phases. In the first phase, the LTEI module integrates six types of learning transfer effects, namely $\mathbf{NTPR}_{c_i \to c_j}$, $\mathbf{NTSR}_{c_i \to c_j}$, and $\mathbf{NTSR}_{\mathcal{N}_{\text{NTSR}}(c_i) \to c_i}$, outputted by the NTE module, and $\mathbf{PTPR}_{c_i \to c_j}$, $\mathbf{PTSR}_{c_i \to c_j}$, and $\mathbf{PTSR}_{\mathcal{N}_{\text{PTSR}}(c_i) \to c_i}$, computed by the PTE module. Then, in the second phase, the LTEI module updates the student's knowledge state in the concept based on the aggregation of all types of learning transfer effects using the gated function $\mathcal{G}$. In the following, we describe the implementation of the LTEI module.

● **Integrating learning transfer effects.** The two-phase integration strategy proposed to aggregate all types of learning transfer effects is as follows:

Phase 1: Aggregating the positive transfer effects or the negative transfer effects for each concept. Given a certain concept c_j which is the successor or neighbor of concept c_i, the aggregation result of the positive transfer prerequisite effect and positive transfer similarity effect at time step t for the concept c_n which can be either c_i or c_j is denoted as $\boldsymbol{A}_{c_n}^{+} \in \mathbf{R}^{d_h}$, while the aggregation result of the negative transfer prerequisite effect and negative transfer similarity effect at time step t for c_n is denoted by $\boldsymbol{A}_{c_n}^{-} \in \mathbf{R}^{d_h}$. $\boldsymbol{A}_{c_n}^{+}$ and $\boldsymbol{A}_{c_n}^{-}$ are calculated using Eqs. (10) and (11), respectively,

$$\boldsymbol{A}_{c_n}^{+} = \begin{cases} \mathbf{PTSR}_{\mathcal{N}_{\text{PTSR}}(c_i) \to c_i}, & c_n = c_i; \\ \alpha \mathbf{PTPR}_{c_i \to c_j} + (1-\alpha)\, \mathbf{PTSR}_{c_i \to c_j}, & c_n = c_j \end{cases} \tag{10}$$

$$\boldsymbol{A}_{c_n}^{-} = \begin{cases} \mathbf{NTSR}_{\mathcal{N}_{\text{NTSR}}(c_i) \to c_i}, & c_n = c_i; \\ \alpha \mathbf{NTPR}_{c_i \to c_j} + (1-\alpha)\, \mathbf{NTSR}_{c_i \to c_j}, & c_n = c_j \end{cases} \tag{11}$$

where α is a hyper-parameter and we set its value to 0.5.

Phase 2: Integrating $\boldsymbol{A}_{c_n}^{+}$ and $\boldsymbol{A}_{c_n}^{-}$. Because different students have different knowledge states at $t-1$ time step, our intuition is that students' sensitivities to positive and negative transfer effects may vary owing to their different knowledge states. Therefore, we use a MultiLayer Perceptron (MLP) to learn the sensitivity value β for each student at each time step, which is used to determine the proportion of positive transfer effects or negative transfer effects when integrating these two types of transfer effects. Specifically, for a certain concept c_n, Eqs. (12)–(14) are used to integrate the aggregated result of all types of positive transfer effects (i.e., $\boldsymbol{A}_{c_n}^{+}$) and the aggregated result of all types of negative transfer effects (i.e., $\boldsymbol{A}_{c_n}^{-}$) at time step t. The final integration result is denoted as $\boldsymbol{I}_{c_n}^{t} \in \mathbf{R}^{d_h}$,

$$\boldsymbol{A}_{c_n} = \beta \cdot \boldsymbol{A}_{c_n}^{+} + (-(1-\beta) \cdot \boldsymbol{A}_{c_n}^{-}) \tag{12}$$

$$\beta = \text{MLP}\left(\sum_{c_n \in N} \boldsymbol{h}_{c_n}^{t-1} \right) \tag{13}$$

$$\boldsymbol{I}_{c_n}^{t} = \tanh(\boldsymbol{W}_{\text{I}} \boldsymbol{A}_{c_n} + \boldsymbol{b}_{\text{I}}) \tag{14}$$

where $\boldsymbol{W}_{\text{I}} \in \mathbf{R}^{d_h \times d_h}$ and $\boldsymbol{b}_{\text{I}} \in \mathbf{R}^{d_h}$ are learnable weight matrix and bias. β is a learnable parameter that serves as a weight for calculation of the weighted sum of $\boldsymbol{A}_{c_n}^{+}$ and $\boldsymbol{A}_{c_n}^{-}$ for each student. Because the positive transfer and negative transfer effect are two opposite effects, it is necessary to add a minus before the negative transfer effect when calculating $\boldsymbol{A}_{c_n}$.

● **Computing knowledge states based on LTE.** Given $\boldsymbol{I}_{c_n}^{t}$, that is, the integration result of the LTE at time step t towards concept c_n, which is calculated based on the learning behaviors of the target student. The LTE-aware knowledge state of the student in concept c_n at time step t, represented as $\tilde{\boldsymbol{h}}_{c_n}^{t} \in \mathbf{R}^{d_h}$, is computed according to

$$\tilde{\boldsymbol{h}}_{c_n}^{t} = \mathcal{G}\left(\left(\boldsymbol{I}_{c_n}^{t} \oplus \boldsymbol{e}(c_n) \right), \boldsymbol{h}_{c_n}^{t-1} \right) \tag{15}$$

where $\mathcal{G}$ is the same GRU gate as Eq. (3) and $\boldsymbol{h}_{c_n}^{t-1}$ denotes the knowledge state of the target student at time step $t-1$. As indicated by Eq. (15), the $\tilde{\boldsymbol{h}}_{c_n}^{t}$ is computed not only by considering the aggregated information in concept c_n (i.e., $\boldsymbol{I}_{c_n}^{t}$) but also by considering the concept's embedding $\boldsymbol{e}(c_n) \in \mathbf{R}^{d_c}$, where d_c denotes the dimensionality of a concept's embedding vector.

4.5 Prediction component

Given a learning transfer graph LTG (VX, EG), the DLE component and LTE component in the proposed LTKT model successively update the knowledge state of the target student for every concept in LTG (VX, EG) based on the direct transfer effect and the learning transfer effect. In the prediction component, the updated knowledge state of the target student for each concept $c_n \in C$ at time step t (i.e., $\tilde{\boldsymbol{h}}_{c_n}^{t}$) is used to predict the probability $\hat{p}_{\text{ex}_{t+1}}^{t+1}$ that the student answers an exercise ex_{t+1} investigating concept $c_m \in C$ correctly at the subsequent time step $t+1$. Specifically, $\hat{p}_{\text{ex}_{t+1}}^{t+1}$ is computed as follows:

$$\begin{gathered} \hat{p}_{\text{ex}_{t+1}}^{t+1} = f_{\text{out}}(\tilde{\boldsymbol{h}}_{c_m}^{t}), \\ f_{\text{out}}(\tilde{\boldsymbol{h}}_{c_m}^{t}) = \sigma(\boldsymbol{W}_{\text{o}} \tilde{\boldsymbol{h}}_{c_m}^{t} + \boldsymbol{b}_{\text{o}}) \end{gathered} \tag{16}$$

where $\boldsymbol{W}_{\text{o}} \in \mathbf{R}^{d_h \times d_h}$ is a learnable weight matrix, $\boldsymbol{b}_{\text{o}} \in \mathbf{R}^{d_h}$ is the learned bias, and $\sigma(\cdot)$ is the activation function sigmoid.

Algorithm 1 shows the comprehensive procedure of the proposed LTKT model.

4.6 Objective function

In the LTKT model, all parameters are jointly optimized by minimizing the standard cross-entropy loss $\mathcal{L}$ between the predicted probability $\hat{p}^t$ and the groundtruth answering result $\mathcal{I}^t$ at each time step t during training,

$$\mathcal{L} = -\sum_{t=1}^{T} (\mathcal{I}^t \log \hat{p}^t + (1 - \mathcal{I}^t) \log (1 - \hat{p}^t)) \tag{17}$$

Note that the binary indicator variable $\mathcal{I}^t$ in Eq. (17) discriminates between correct ($\mathcal{I}^t = 1$) and incorrect ($\mathcal{I}^t = 0$) responses to the exercise at time t.

The objective function is minimized in mini-batches. Further elaboration on the training specifics can be found in Section 5.

5 Experiment

In this part, we begin by discussing the experimental settings and then conduct a comprehensive analysis of the experimental results.

5.1 Experimental settings

5.1.1 Datasets

To assess the efficacy of the proposed LTKT model, two publicly available datasets are employed for evaluation. The distribution of students' interaction sequence lengths in each dataset is depicted in Fig. 3. Brief descriptions of the two datasets are provided below.

- **ASSISTments 2015 (ASSIST15)[§]:** it is collected from the online tutoring platform ASSISTment and comprises learner interaction records from the year 2015. ASSIST15 is a "Skill Builder" dataset, i.e., a student is considered to have mastered a concept when a certain criterion (normally set to answered 3 exercises correctly in a row) is met, and no more exercises will be given after mastery. In our experiment, the preprocessed dataset provided by Zhang et al.[13] is used. Among all the datasets, this dataset exhibits the shortest average sequence length, as shown in Fig. 3.

[§]https://sites.google.com/site/assistmentsdata/home/2015-assistments-skillbuilder-data

Algorithm 1 LTKT Model.

Input: Learning transfer graph LTG (V, E); target student's learning interaction vector $\tilde{\boldsymbol{x}}_t \in \{0,1\}^{2N}$; concept currently being learned $c_i \in C$; neighborhood function $\mathcal{N}_{\text{PTSR}}$ and $\mathcal{N}_{\text{NTSR}}$; successor-hood function $\mathcal{S}_{\text{PTPR}}$ and $\mathcal{S}_{\text{NTPR}}$; target student's knowledge state for N concepts at time step $t-1$, i.e., $\boldsymbol{y}_{t-1} = \left\{\tilde{\boldsymbol{h}}_{c_1}^{t-1}, \tilde{\boldsymbol{h}}_{c_2}^{t-1}, \ldots, \tilde{\boldsymbol{h}}_{c_n}^{t-1}, \ldots, \tilde{\boldsymbol{h}}_{c_N}^{t-1}\right\}$, with $\tilde{\boldsymbol{h}}^{t-1} \in \mathbf{R}^{d_h}$.

Output: Target student's knowledge state for N concepts at time step t, i.e., $\boldsymbol{y}_t = \left\{\tilde{\boldsymbol{h}}_{c_1}^{t}, \tilde{\boldsymbol{h}}_{c_2}^{t}, \ldots, \tilde{\boldsymbol{h}}_{c_n}^{t}, \ldots, \tilde{\boldsymbol{h}}_{c_N}^{t}\right\}$, with $\tilde{\boldsymbol{h}}^{t} \in \mathbf{R}^{d_h}$; predicted answering result of the target student in exercise ex_{t+1} at time step $t+1$, i.e., $\hat{p}_{\text{ex}_{t+1}}^{t+1}$.

$\tilde{\boldsymbol{h}}_{c_i}^{t} = \mathcal{G}\left(\tilde{\boldsymbol{x}}_t \boldsymbol{M}_x, \tilde{\boldsymbol{h}}_{c_i}^{t-1}\right)$; // Modeling the direct learning effect (Eq. (3))
for c_j in $\mathcal{S}_{\text{PTPR}}(c_i)$ **do** // Modeling the positive transfer prerequisite effect (Eq. (4))
    $\mathbf{PTPR}_{c_i \to c_j} = \text{RELU}\left(\boldsymbol{W}_{\text{pp}} \boldsymbol{Q}_{c_i \to c_j} + \boldsymbol{b}_{\text{pp}}\right)$;
end for
for c_j in $\mathcal{N}_{\text{PTSR}}(c_i)$ **do** // Modeling the positive transfer similarity effect (Eqs. (5) and (6))
    $\mathbf{PTSR}_{c_i \to c_j} = \text{RELU}\left(\boldsymbol{W}_{\text{ps}} \boldsymbol{U}_{c_i \to c_j} + \boldsymbol{b}_{\text{ps}}\right)$;
    // From c_i to its neighbor c_j
end for
$\mathbf{PTSR}_{\mathcal{N}_{\text{PTSR}}(c_i) \to c_i} = \sum_{c_j \in \mathcal{N}_{\text{PTSR}}(c_i)} \mathbf{PTSR}_{c_i \to c_j}$;
    // From c_i's neighbors to c_i
for c_j in $\mathcal{S}_{\text{NTPR}}(c_i)$ **do** // Modeling the negative transfer prerequisite effect (Eq. (7))
    $\mathbf{NTPR}_{c_i \to c_j} = \text{RELU}\left(\boldsymbol{W}_{\text{np}} \boldsymbol{R}_{c_i \to c_j} + \boldsymbol{b}_{\text{np}}\right)$;
end for
for c_j in $\mathcal{N}_{\text{NTSR}}(c_i)$ **do** // Modeling the negative transfer similarity effect (Eqs. (8) and (9))
    $\mathbf{NTSR}_{c_i \to c_j} = \text{RELU}\left(\boldsymbol{W}_{\text{ns}} \boldsymbol{D}_{c_i \to c_j} + \boldsymbol{b}_{\text{ns}}\right)$;
    // From c_i to its neighbor c_j
end for
$\mathbf{NTSR}_{\mathcal{N}_{\text{NTSR}}(c_i) \to c_i} = \sum_{c_j \in \mathcal{N}_{\text{NTSR}}(c_i)} \mathbf{NTSR}_{c_i \to c_j}$;
    // From c_i's neighbors to c_i
Use a two-phase integration strategy (Eqs. (10)−(14)) to derive the integration result of learning transfer effects $\boldsymbol{I}_{c_n}^{t} \in \mathbf{R}^{d_h}$ for each concept c_n; // Aggregating of the learning transfer effects
$\tilde{\boldsymbol{h}}_{c_n}^{t} = \mathcal{G}\left(\left(\boldsymbol{I}_{c_n}^{t} \oplus \boldsymbol{e}(c_n)\right), \tilde{\boldsymbol{h}}_{c_n}^{t-1}\right)$; // Updating knowledge state on c_n based on the learning transfer effects and obtaining $\boldsymbol{y}_t$ (Eq. (15))
$\hat{p}_{\text{ex}_{t+1}}^{t+1} = \sigma\left(\boldsymbol{W}_{\text{o}} \tilde{\boldsymbol{h}}_{c_m}^{t} + \boldsymbol{b}_{\text{o}}\right)$; // Predicting the target student's performance on the next exercise ex_{t+1} investigating concept c_m at time step $t+1$ (Eq. (16))
return $\boldsymbol{y}_t$ and $\hat{p}_{\text{ex}_{t+1}}^{t+1}$.

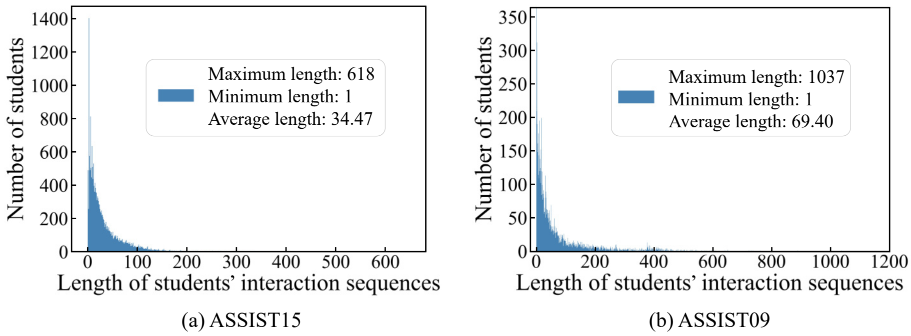


Fig. 3 Distribution of the length of students' interaction sequences in each dataset.

- **ASSISTments 2009 (ASSIST09)¶:** it is also derived from the ASSISTments online tutoring system but with respect to the years from 2009 to 2010. This dataset has been extensively used by several researchers to evaluate the KT models. In specific, ASSIST09 contains two sub-datasets, namely "Skill-Builder data 2009–2010" and "Non-Skill Builder data 2009–2010". The "Non-Skill Builder data 2009–2010" is used in our experiments, where students are asked to complete the exercises in a predetermined orders, randomized order, or some other orders. Following the setting in Ref. [6], only the first investigated pedagogical concept of an exercise is used to label the exercise in the experiments. In addition, we remove all records with correctness labels not equal to 1 or 0.

Table 3 provides an overview of the statistical characteristics of the aforementioned datasets. The value of "Proportion of NTSR" in Table 3 is derived using the function $\frac{\text{Number of NTPR}}{N\times(N-1)}$ and the value of "Proportion of NTSR" is computed as $\frac{\text{Nubmer of NTSR}}{N\times(N-1)}$. Here, $N\times(N-1)$ represents the total number of relations in a directed complete graph, where N denotes the number of concepts (or vertices) in the graph. As shown in Table 3, the two types of negative transfer relations (i.e., NTPR and NTSR) account for a higher portion of the ASSIST15 dataset than the ASSIST09 dataset.

¶https://sites.google.com/site/assistmentsdata/home/2009-2010-assistment-data/non-skill-builder-data-2009-10

5.1.2 Learning transfer graph construction

As none of the datasets provides an explicit learning transfer graph LTG (VX, EG), we determine the vertex set VX and edge set EG of LTG (VX, EG) as follows. First, all the pedagogical concepts of the course are used to generate the vertex set VX of the LTG (VX, EG), where each vertex is associated with a different concept. Then, inspired from previous works[15, 16, 32], effective strategies are designed to determine four types of transfer relations (i.e., PTPR, PTSR, NTPR, NTSR) which have been explained in Section 3.1, and to derive the edge set EG for the LTG (VX, EG).

To facilitate the determination of the PTPR and PTSR relations among vertices in LTG (VX, EG), a positive $N\times N$ transfer statistical matrix $\boldsymbol{P}$ is computed, where every element $\boldsymbol{P}[i][j]\in\boldsymbol{P}$ records the occurrence frequency of the event "$i\checkmark\to j\checkmark$". Notice that $\boldsymbol{P}[i][j]$ is set to 0 when $i=j$. Meanwhile, for the convenience of mining the NTPR and NTSR relations among vertices in LTG (VX, EG), a negative $N\times N$ transfer statistical matrix $\boldsymbol{L}$ is calculated. Similarly, the value of all $\boldsymbol{L}[i][i]$ with $i\leqslant N$ in the matrix are consistently set to 0. For the case that $i\neq j$, the value of $\boldsymbol{L}[i][j]$ is the summation of two statistical values, i.e., the occurrence frequency of the event "$i\times\to j\checkmark$" and the occurrence frequency of the event "$i\checkmark\to j\times$". The distribution of the interaction events in Figs. 4 and 5 show that there are enough events having the form of "$i\checkmark\to j\checkmark$", "$i\checkmark\to j\times$", and "$i\times\to j\checkmark$" in students' historical learning log. In this case, different types of transfer

Table 3 Statistic of all dataset.

Dataset	Number of pedagogical concepts	Number of learners	Number of interactions	Average length of students' interaction sequences	Number of vertices (N)	Number of PTPRs	Number of PTSRs	Number of NTPRs	Number of NTSRs	Proportion of NTPR in all relations	Proportion of NTSR in all relations
ASSIST15	100	19 840	683 801	34.47	100	1112	1384	1203	1394	12.15%	14.08%
ASSIST09	154	8026	557 030	69.40	154	2587	2152	2692	2796	11.43%	11.87%

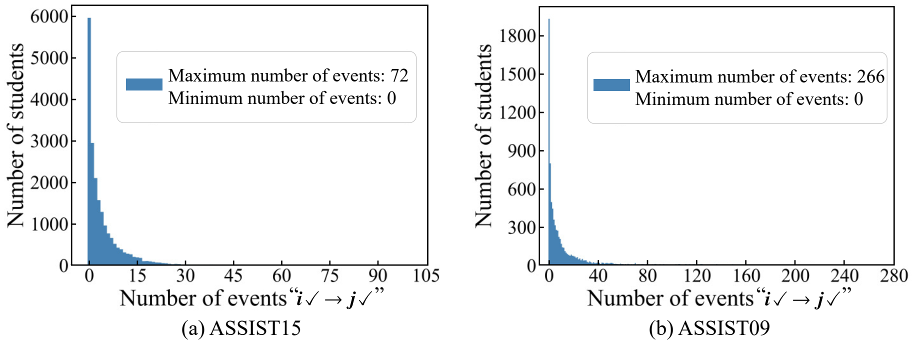


Fig. 4 Distribution of the number of interaction events "$i\checkmark \to j\checkmark$" in each dataset.

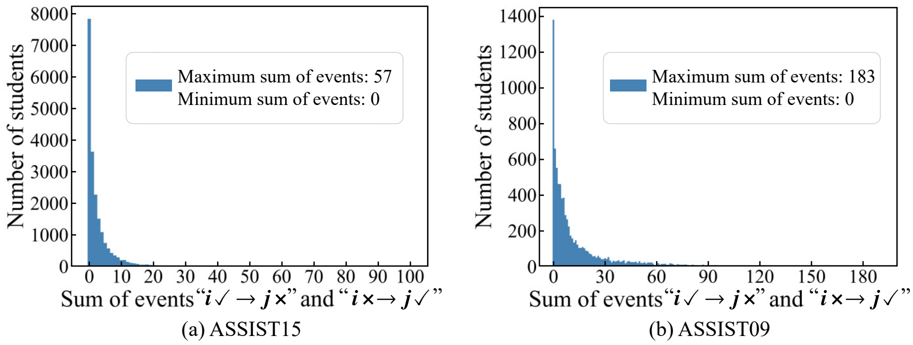


Fig. 5 Distribution of the total amount of the two interaction events "$i\checkmark \to j\times$" and "$i\times \to j\checkmark$" in each dataset.

relations can be determined based on the following statistical strategies:

- **Determining PTPR**. To determine the PTPR among vertices, a non-symmetrical matrix called $\mathbf{PT} \in \mathbf{R}^{N\times N}$ is computed to record the probabilities of a PTPR existing between every pair of concepts. Specifically, when $i \neq j$, the equation $\mathbf{PT}[i][j] = \dfrac{\boldsymbol{P}[i][j]}{\sum_k \boldsymbol{P}[i][k]}$ is used to calculate the probability that the knowledge of concept c_i is positively transfered to concept c_j unidirectionally. Here, k is an index variable used to iterate over the N concepts. Subsequently, a threshold θ_{PTPR} is introduced to determine whether there is a PTPR between a concept to another concept, i.e., $\boldsymbol{P}[i][j] \geqslant \theta_{\text{PTPR}}$ indicating a PTPR relation from concept c_i to concept c_j. The threshold is set as the average value of all elements in $\mathbf{PT}$, i.e., $\theta_{\text{PTPR}} = \dfrac{\sum_{i,j}\mathbf{PT}[i][j]}{N\times N}$.

- **Determining PTSR**. To determine the PTSR among vertices, a symmetric matrix called $\mathbf{PS} \in \mathbf{R}^{N\times N}$ is computed to record the probabilities of a PTSR existing between every pair of concepts. To compute $\mathbf{PS}$, the equation $\mathbf{FQPS}[i][j] = \dfrac{\boldsymbol{P}[i][j] + \boldsymbol{P}[j][i]}{|\boldsymbol{P}[i][j] - \boldsymbol{P}[j][i]| + \epsilon}$ is used to calculate the frequency of concurrent occurrences between two concepts, and when both "$i\checkmark \to j\checkmark$" and "$j\checkmark \to i\checkmark$" occur frequently and frequency gap is small, the frequency of concurrent occurrences is higher, where $\epsilon = 0.1$ is a small value set to prevent zero division. Then, each element in $\mathbf{FQPS}$ is normalized to the range of $[0,1]$ using the max-min-scaling operation, i.e., $\mathbf{PS}[i][j] = \dfrac{\mathbf{FQPS}[i][j] - \min(\mathbf{FQPS})}{\max(\mathbf{FQPS}) - \min(\mathbf{FQPS})}$. Apparently, $\mathbf{PS}[i][j] \in [0,1]$ is the probability that a PTSR exists between concept i and concept j bidirectionally. Then, a pair of concepts (c_i, c_j), whose $\mathbf{PS}[i][j]$ is larger than or equal to a threshold θ_{PTSR}, is deemed connected by an edge of PTSR type in LTG (VX, EG). Here, the threshold θ_{PTSR} is set to the average value of all elements in $\mathbf{PS}$, i.e., $\theta_{\text{PTSR}} = \dfrac{\sum_{i,j}\mathbf{PS}[i][j]}{N\times N}$.

- **Determining NTPR**. To determine the NTPR among vertices in LTG (VX, EG), a non-symmetrical matrix denoted as $\mathbf{NT}$ is computed. When $i \neq j$, we use $\mathbf{NT}[i][j] = \dfrac{\boldsymbol{L}[i][j]}{\sum_k \boldsymbol{L}[i][k]}$ to calculate the probability of a negative transfer prerequisite relation between the two concepts. Here, $\boldsymbol{L}$ is the negative transfer count graph. $\mathbf{NT}[i][j]$ indicates the probability that the knowledge of concept i is negatively transfered to concept j unidirectionally. Subsequently, we set a threshold θ_{NTPR} to determine whether there is a relation between a concept pair (i,j), i.e., if $\mathbf{NT}[i][j] \geqslant \theta_{\text{NTPR}}$, then $\mathbf{NT}[i][j]=1$. Where θ_{NTPR} is set to the average value of all elements in **NT**, i.e., $\theta_{\text{NTPR}} = \dfrac{\sum_{i,j}\mathbf{NT}[i][j]}{N\times N}$.

• **Determining NTSR**. To determine the NTSR among vertices in LTG (VX, EG), a symmetric matrix denoted as $\mathbf{NS}\in \mathbf{R}^{N\times N}$ is computed. In specific, we first compute a frequency of concurrent occurrences for every $\mathbf{NS}[i][j]\in \mathbf{NS}(i\neq j)$, which is denoted by $\mathbf{FQNS}[i][j]$, based on the equation $\mathbf{FQNS}[i][j] = \dfrac{N[i][j]+N[j][i]}{|N[i][j]-N[j][i]|+\epsilon}$. Here, we set ϵ to a small value of 0.1 in order to prevent division by zero. Subsequently, the max-min-scaling technique is employed to normalize every $\mathbf{FQNS}[i][j]$ to the range of $[0,1]$, so that to derive $\mathbf{NS}[i][j]\in\mathbf{NS}$, i.e., $\mathbf{NS}[i][j] = \dfrac{\mathbf{FQNS}[i][j]-\min(\mathbf{FQNS})}{\max(\mathbf{FQNS})-\min(\mathbf{FQNS})}$. Then, $\mathbf{NS}[i][j]$ is the computed probability of negative transfer between concept i and concept j bidirectionally. Finally, we also set a threshold θ_{NTSR} towards each $\mathbf{NS}[i][j]\in\mathbf{NS}$ to determine whether a pair of concepts (c_i,c_j) is connected by the edge of NTSR type in LTG (VX, EG), where θ_{NTSR} is set as the average of all elements in **NS**, i.e., $\theta_{\text{NTSR}} = \dfrac{\sum_{i,j}\mathbf{NS}[i][j]}{N\times N}$.

Note that when two concepts c_i and c_j are not linked by a relation, their corresponding statistics (i.e., $\mathbf{PT}[i][j]$, $\mathbf{PS}[i][j]$, $\mathbf{NT}[i][j]$, and $\mathbf{NS}[i][j]$) are set to a very small value γ rather than 0. This setting is inspired from Ref. [6], which shows that simply setting to $\gamma=0$ cannot ensure the effectiveness of KT based on the knowledge structure. Furthermore, the threshold used to determine the existence of a certain type of relation (i.e., θ_{PTPR}, θ_{PTSR}, θ_{NTPR}, or θ_{NTSR}) is set following the setting in Ref. [16]. Then, the edge set E of LTG (VX, EG) can be obtained by merging four types of edges: PTPR, PTST, NTPR, and NTSR.

5.1.3 Baselines

The effectiveness of LTKT is assessed through a comparative analysis against a group of baseline models. Specifically, we compare four deep learning-based KT models that fail to consider learning transfer effects, i.e., DKT, regularized Deep Knowledge Tracing (DKT+), DKVMN, and CKT. In addition, four deep learning-based KT models that only consider positive learning transfer among pedagogical concepts are used as baselines, i.e., ContextKT, SPARSEKT, GKT, and SKT. Details of each baseline are discussed below.

• **DKT☆:** DKT[12] stands as a pioneering KT model that introduces the concept of deep learning. It leverages a Recurrent Neural Network (RNN) or Long Short-Term Memory (LSTM) network to model the evolving knowledge states of a student. The input of DKT is the one-hot vector representation of each student's interaction sequence. Upon the completion of input processing by the RNN or LSTM, the hidden state of the network is interpreted as the target student's knowledge state.

• **DKT+©:** DKT+[24] is an extended KT model based on DKT and aims to tackle two fundamental problems of DKT. The first problem of DKT is that it fails to reconstruct the observed inputs. Second, in DKT, the predicted performance of a student in knowledge concepts across different time steps is not consistent. To address these two problems, DKT+ incorporates three regularization terms into the loss function of the original DKT model, aiming to enhance the consistency in prediction. In DKT+, the loss function is defined as $\mathcal{L}' = \mathcal{L}+\lambda_r r+\lambda_{w_1}w_1+\lambda_{w_2}w_2^2$, where r is the regularization term for the reconstruction problem; two waviness measures w_1 and w_2 are regularization terms that smoothen the transition in the prediction; λ_r, λ_{w_1}, and λ_{w_2} are regularization parameters.

• **DKVMN☆:** DKVMN[13] is another classical KT model inspired by MANN. Specifically, in DKVMN, a static key matrix is utilized to retain latent concepts, while a dynamic value matrix is employed for storing and updating a student's knowledge states in various concepts through read and write operations.

• **CKT¤:** CKT[14] is the first KT model to introduces CNN into the domain of KT. It utilizes students' historical learning interactions to measure their individualized prior knowledge and employs hierarchical convolutional layers to extract individualized learning rates based on continuous learning interactions. The output of the hierarchical convolutional layers in CKT represents the knowledge state of the target student. Finally, the dot product between the student's current knowledge state and the embedding of the next exercise is leveraged to predict the student's prospective performance.

• **ContextKT:** ContextKT[37] leverages an LSTM to capture and represent the change of each student's

☆https://github.com/bigdata-ustc/XKT
©https://github.com/ckyeungac/deep-knowledge-tracing-plus
¤https://github.com/bigdata-ustc/Convolutional-Knowledge-Tracing

knowledge state and employs an attention mechanism to derive the total positive transfer impact of similar concepts investigated by historical exercises (historical similar concepts for short) on the currently examined concept (current concept for short). Finally, the knowledge state of the current concept and the derived total positive transfer impact of historical similar concepts on the current concept are concurrently used to predict a student's performance on next exercise.

- **SPARSEKT©:** SPARSEKT[39] is a deep model built based on an attention mechanism that considers the positive transfer between historical similar concepts and the current concept. Moreover, in order to enhance the robustness and generalization of the attention based deep learning KT model, SPARSEKT proposes two sparsification heuristics, i.e., the soft-thresholding sparse attention and the top-K sparse attention.
- **GKT☆:** GKT[15] reformulates the KT task as a time-series node-level classification problem in GNN. Although GKT considers the learning transfer effects and updates the knowledge state of the target student based on the knowledge structure, it only considers the PTPR between concepts when constructing the knowledge structure. In other words, GKT ignores the negative transfer in modelling learning transfer. In the experiment, we set the size of all hidden vectors and embedded matrices in GKT to 32.
- **SKT☆:** SKT[16] is another prominent KT model that takes into account the learning transfer effects. It models the influence propagation between concepts using both PTPR and PTSR in the knowledge structure. However, SKT only models positive transfer effects while negative transfer effects are ignored. In the experiment, we set the size of all hidden vectors and embedded matrices in SKT to 64.

©https://pykt.org/

Table 4 provides an overview of the characteristics of each comparison model and clearly highlights the differences between our proposal and all comparison baselines.

5.1.4 Evaluation metrics

Inspired by previous studies[12, 13], we evaluate the proposed LTKT model from a classification perspective. During the evaluation, to convert the predicted probability, denoted as $\hat{p}^{t+1}$, representing the probability that the target student can correctly answer the next exercise at time step $t+1$, into a binary value of either 1 or 0, we employ a threshold of 0.5, where $\hat{p}^{t+1} > 0.5$, corresponding to a value of 1, it signifies that the student successfully answers the exercise, indicating a positive sample. Conversely, when $\hat{p}^{t+1} \leqslant 0.5$, denoted as 0, it indicates that the student answers the exercise incorrectly, representing a negative sample. Subsequently, in the following experiments, two extensively utilized evaluation metrics, specifically the Area Under Receiver Operating Characteristic (ROC) Curve (AUC) and F1-Score, are employed as evaluation metrics. Specifically, for an ROC curve generally lying above the line $y=x$, the AUC metric generally takes a value in the range [0.5, 1]. In particular, an AUC value of 0.5 indicates that the classification performance is equivalent to random guessing, and a higher AUC value indicates better classification performance. The F1-Score which takes a value in the range of [0, 1] takes into account both the precision and recall to evaluate a classification model. A value of 1 signifies the model's optimal output, while 0 indicates the poorest model performance.

5.1.5 Other settings

- **Data processing:** For each dataset, the interaction sequences of students with only one interaction are first removed, after which all remaining sequences are

Table 4 Characteristic of the comparison method (×: aspect excluded; √: aspect included).

Model	Learning transfer	PTPR	PTSR	NTPR	NTSR	Network
DKT[12]	×	×	×	×	×	RNN
DKVMN[13]	×	×	×	×	×	Key-value memory
CKT[14]	×	×	×	×	×	CNN
GKT[15]	√	√	×	×	×	GNN+RNN
SKT[16]	√	√	√	×	×	GNN+RNN
DKT+ [24]	×	×	×	×	×	RNN
ContextKT[37]	√	×	×	×	×	RNN+attention networks
SPARSEKT[39]	√	×	×	×	×	Attention networks
LTKT (ours)	√	√	√	√	√	GNN+RNN

consistently set to a fix length of 200. Specifically, we split a sequence into several sub-sequences if the sequence is longer than the fix length, and in cases where a sequence falls short of a predetermined fixed length, we utilize a zero vector for padding, thereby ensuring the sequence conforms to the desired length. In addition, each dataset is divided into training and test sets, maintaining an 8:2 ratio, taking into account the cardinality of students, and 20% of the samples are extracted from the training set to form the validation set. It should be noted that in the experiments, the learning transfer graph LTG (VX, EG) is constructed using all samples from the training set, test set, and validation set.

● **Framework setting:** For all dataset, the dimensions d_v and d_c of those embedding matrices are consistently set to 32, while the dimensionality of the knowledge state vector $\boldsymbol{h}$, i.e., d_h, is set to 32. Inspired by Ref. [16], we set α as 0.5 for all datasets in Eqs. (10) and (11). Inspired by Ref. [6], we set γ to 0.05 for all datasets. The hyperparameter analysis of γ is provided in Section 5.4.

● **Training details:** In order to keep the scale of the gradients roughly the same in all layers of a deep neural network, we apply Xavier initialization[43] to initialize the parameters of all layers. To minimize the objective function, we utilize the Adam optimizer[44]. Note that all parameters are learned during the training stage and the model that achieves the highest performance on the validation set is selected to evaluate on the test set and generate the final reported results. We establish the initial learning rate at 0.001, set the maximum number of training epochs to 30, and employ mini-batches of size 8 for all datasets. Each of the compared KT models is trained and tested using i9 9900 64G CPU and RTX 2080 8G GPU.

5.2 Comparison with baselines

Within this section, we compare the LTKT with its baselines based on their performance in handling student performance prediction task, which serves as a widely recognized yardstick for assessing the overall effectiveness of KT models[5, 6, 12, 13, 16]. The comparison results with respect to the two metrics, namely AUC and F1-Score, are presented in Table 5. Based on the insights gained from the results presented in Table 5, enabling us to derive noteworthy conclusions and pivotal observations.

Finding 1: Considering the investigation relations between exercises and concepts is helpful in improving the effectiveness of KT models. This finding is derived by comparing DKT or DKT+ with DKVMN in Table 5. The DKVMN, which considers the relations between exercises and concepts, gains an average of 5.18% improvement in AUC and 9.66% improvement in F1-Score on all datasets compared to DKT+, which fails to consider such investigation relations when tracing students' knowledge states.

Finding 2: Considering the knowledge structure (i.e., learning transfer relations among concepts) is important in building KT models. This discovery stems from two observations from Table 5. First, ContextKT and SPARSEKT perform better than DKT, which indicates that considering the learning transfer relations between historical similar concepts and current concept is of importance in optimizing the performance of a KT task. Second, GKT and SKT consistently outperform the remaining baselines. Since the GKT and SKT use learning transfer among concepts to enhance their KT performance, we can

Table 5 Comparison with baselines. The best results are in bold and the second best results are underlined.

Model	ASSIST15		ASSIST09	
	AUC	F1-Score	AUC	F1-Score
DKT[12]	0.659277	0.474262	0.701398	0.644576
DKVMN[13]	0.704222	0.518766	0.753741	0.686000
CKT[14]	0.705449	0.482538	0.725866	0.664279
GKT[15]	0.704705	0.501025	<u>0.766937</u>	<u>0.694213</u>
SKT[16]	<u>0.720029</u>	<u>0.531970</u>	0.753428	0.684633
DKT+ [24]	0.680255	0.475407	0.705510	0.622474
ContextKT[37]	0.701648	0.447117	0.759885	0.686735
SPARSEKT[39]	0.701073	0.482143	0.753048	0.675488
LTKT (ours)	**0.734894**	**0.591427**	**0.774714**	**0.706619**
Improvement (with respect to the second best)	2.0645%	11.1767%	1.014%	1.787%

conclude that learning transfer based on knowledge structure extracting from all students interaction logs is an indispensable factor in ensuring the quality of KT tasks.

Finding 3: Concurrently considering the positive and negative learning transfers is important for KT. This finding is derived by comparing ContextKT, SPARSEKT, GKT, and SKT with the proposed LTKT. As listed in Table 5, although these four baselines consider the learning transfer effect based on knowledge structure, they focus on the effect of positive learning transfer while ignoring the impact of negative learning transfer. Therefore, these four baselines can only generate less satisfactory results, while LTKT outperforms each of these four baselines on all datasets. In particular, LTKT gains at most 2.06% in AUC and 11.18% in F1-Score improvements on the ASSIST15 dataset, compared to the second-best SKT method. Therefore, we can conclude that considering both positive and negative learning transfers is indispensable for ensuring the quality of KT results.

Finding 4: Fully exploring and exploiting relations among concepts is vital in ensuing the robustness of KT tasks when dealing with students who have limited historical interactions. Table 3 shows that the ASSIST15 dataset has the smallest average length of students' interaction sequences compared to the remaining datasets. Analyzing the results presented in Table 5, the proposed LTKT model still performs very well on the ASSIST15 dataset compared to all baselines when the average length of students' interaction sequences is 34.47. On the other hand, the performance of GKT encounters a significant decline on the ASSIST15 dataset because it ignores not only the effect of negative learning transfer but also the similarity relation between two pedagogical concepts. This, once again, verifies the importance of fully exploring and exploiting relations among concepts to guarantee the performance of KT when coping with students with limited historical interactions.

Finding 5: Proposed LTKT model performs better when the proportion of NTPR or NTSR in the dataset is larger. As shown in Table 3, the proportion of NTPR (or NTSR) in the ASSIST15 dataset is apparently greater than the proportion of NTPR (or NTSR) in the ASSIST09 dataset. In this case, the experimental results presented in Table 5 clearly indicate that the improvements gained by LTKT compared with the next-best method in the ASSIST15 dataset are significantly larger than those gained in the ASSIST09 dataset. This observation verifies the effectiveness of LTKT in dealing with the negative transfer relations among pedagogical concepts, which are crucial for enhancing the KT performance of existing KT models.

5.3 Ablation study

Within this portion, we conduct an ablation experiment to evaluate the individual effectiveness of each proposed module in LTKT. The results of this ablation study, comparing four different variants of LTKT, are presented in Table 6. The implementations of the four variants of LTKT are elaborated below.

- **LTKT-LTE:** This is a variant of LTKT in which the LTE component is removed. Note that, without the LTE component, only the direct learning effect is considered by the variant.
- **LTKT-NTE:** This is a variant of LTKT in which the NTE module in the LTE component is deleted from LTKT. In this case, LTKT-NTE does not consider the negative learning transfer relations among pedagogical concepts when modeling the knowledge structure.
- **LTKT-PTE:** This is a variant of LTKT in which the PTE module in the LTE component is deleted from LTKT. Therefore, LTKT-PTE fails to model and utilize information of positive learning transfers in KT.
- **LTKT-β:** This is a variant of LTKT. In LTKT-β,

Table 6 Result of ablation study. The best results are in bold. "√" represents that the model considers the effect, and "×" indicates the model does not consider that effect.

KT model	DL	PTE	NTE	β	ASSIST15		ASSIST09	
					AUC	F1-Score	AUC	F1-Score
LTKT-LTE	√	×	×	×	0.703749	0.512215	0.710134	0.633324
LTKT-NTE	√	√	×	×	0.729341	0.573903	0.767337	0.698561
LTKT-PTE	√	×	√	×	0.732710	0.569713	0.766737	0.697373
LTKT-β	√	√	√	×	0.734723	0.578351	0.769226	0.702618
LTKT (ours)	√	√	√	√	**0.734894**	**0.591427**	**0.774714**	**0.706619**

the learnable hyperparameter $\beta \in (0,1)$, which is utilized to determine the proportion of the positive transfer effect (i.e., β) and the proportion of the negative transfer effect (i.e., $1-\beta$) when integrating these two types of effects, is eliminated from LTKT. Instead, LTKT-β simply adds up these two types of learning transfer effects together.

From Table 6, we first observe that the LTE component contributes significantly to the LTKT performance. Specifically, without the LTE component, the AUC of LTKT-LTE decreases by an average of 6.29% and the F1-Score declines averagely by 11.88% on the two datasets compared with LTKT. This observation, once again, verifies the importance of considering learning transfer effects among pedagogical concepts when tracing students' knowledge states. Another observation from Table 6 is that either the positive or negative transfer effect plays a crucial role in ensuring the performance of KT models. This observation is derived by comparing LTKT-NTE (or LTKT-PTE) with LTKT. As shown in Table 6, compared with LTKT, LTKT-NTE, which closes the NTE module performs on average 1.4528% worse for all metrics on all datasets, while LTKT-PTE, which removes the PTE performs on average 1.5766% worse than LTKT on all datasets. Finally, by making a comparison of LTKT and LTKT-β we can conclude that modeling students' individual sensitivity to positive and negative transfer effects can improve the performance of LTKT model. This is due to the fact that, for all datasets, LTKT that models students' individual sensitivity to positive and negative transfer effects performs 0.8917% on average, better than its variant LTKT-β, which simply adds up two types of transfer effects.

5.4 Hyperparameter analysis

Inspired by Ref. [6], in the proposed LTKT model, we use a very small value γ to compensate for possible omissions or mistakes in the learning transfer graph. To evaluate how different γ values affect the performance of LTKT, experiments are conducted by separately setting γ to each of these values in the set {0, 0.01, 0.03, 0.05, 0.07, 0.09}, and listing the performance of LTKT in Fig. 6. Figure 6 shows that with the growth of γ, the performance of LTKT in terms of AUC (or F1-Score) initially increases and the best performance is achieved when $\gamma = 0.05$ on all datasets. However, when γ is larger than 0.05, the performance of the LTKT degrades rapidly for all datasets.

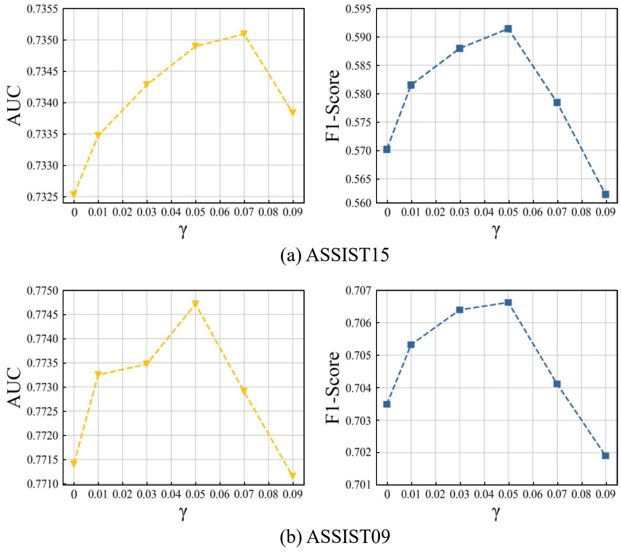
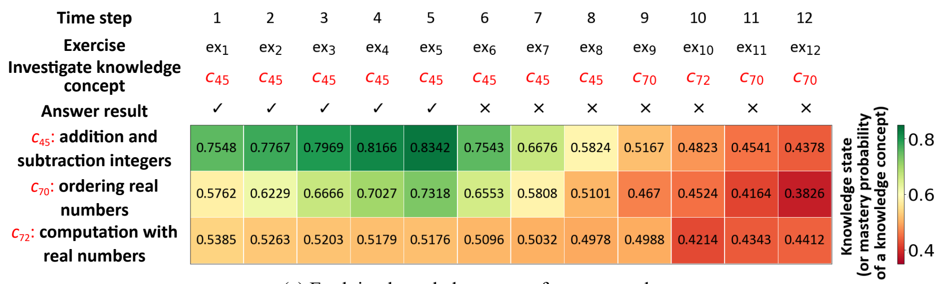


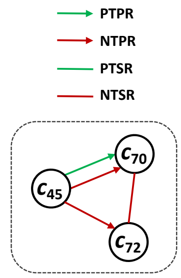
Fig. 6 Impact of γ on LTKT.

5.5 Visualization of knowledge state evolution

The purpose of this section is to evaluate the effectiveness of proposed LTKT in accurately capturing the evolution of a student's knowledge states over time. Specifically, Fig. 7a illustrates the evolving knowledge states of the target student s_1 on ASSIST09 dataset which are predicted using LTKT. In Fig. 7a, every row records the evolving knowledge states of a concept over 12 time steps, and the learning transfer graph LTG (VX, EG) modeling the relations among three concepts in the figure is displayed in Fig. 7b. Note that Fig. 7a is a heat map, where a warm color indicates a relatively low proficiency value for knowledge and a cold color represents a relatively large proficiency value for knowledge. As shown in Fig. 7a, from time step 1 to 5, student s_1 correctly answered 5 consecutive exercises that investigate the same pedagogical concept c_{45} (addition and subtraction integers), which makes student s_1's proficiency value in concept c_{45} achieve continuous improvement. In the meanwhile, the proficiency value of concept c_{72} (computation with real numbers), which is the successor concept of concept c_{45} in a NTPR, as shown in Fig. 7b, suffers a continuous decline within the same time interval. We believe that the reason for this phenomenon is that fractional arithmetic is not investigated by concept c_{45}, but is emphatically inspected by concept c_{72}. In this case, when the target student is asked to alternately calculate the equations



(a) Evolving knowledge states of a target student



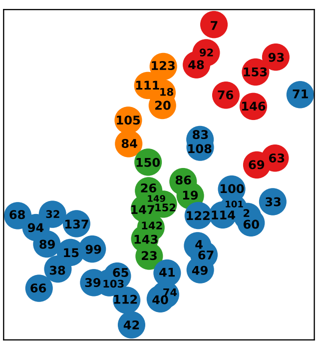
(b) Learning transfer graph

Fig. 7 Example of a target student s_1's evolving knowledge states with respect to 3 concepts (i.e., c_{45}, c_{70}, and c_{72}) when student s_1 is asked to solve 12 exercises (i.e., ex_1, …, ex_{12}) extracted from the ASSIST09 dataset. "×" indicates that s_1 answered the question incorrectly, while "√" denotes a correct answer from s_1.

with integers (i.e., c_{45}) and real numbers (i.e., c_{72}), the student may be confused about the processing of the fractional part of the equation and produce wrong computation results. For instance, such confusion may lead to omission of the fractional part or overlooking the decimal point. Figure 7a also shows that student s_1's proficiency in concept c_{70} (i.e., ordering real numbers) increases from time step 1 to 5, along with his/her proficiency in concept c_{45}. Based on the learning transfer graph shown in Fig. 7b, we conclude that although concept c_{70} is both a successor of concept c_{45} in PTPR and a successor of concept c_{45} in a NTPR, the impact of the PTPR is greater than that of the NTPR in these time steps. This may be because learning concept c_{45} helps the student master basic mathematical operations, which then provides a strong foundation for understanding higher-order concepts like c_{70}, i.e., ordering real numbers. In summary, the visualization results of the target student's knowledge state evolution generated by the proposed LTKT model show LTKT has good explanatory power for tracing students' knowledge states.

5.6 Concepts clustering

The embedding results of pedagogical concepts are initialized in LTKT using Xavier initialization[16]. Due to its high interpretability and accuracy in tracing knowledge states, the LTKT model not only effectively trace students' knowledge states but also produces meaningful embeddings for each concept after training. In Fig. 8, we visualize the embeddings of 55 randomly selected concepts from the dataset ASSIST09 using the T-SNE method[45]. As illustrated in Fig. 8, four different colors are used to demonstrate concepts from four different knowledge domains. Figure 8 shows that concepts related to arithmetic (i.e., concepts 89, 15, 99, and 137) are gathered together, concepts related to geometry (i.e., concepts 23, 143, 142, and 147) are clustered together, and concepts related to algebra (i.e., concepts 105, 20, 18, and 111) are distributed near



(a) Clustering result of concepts. Each circle represents a concept, with the centered number indicating its ID. Concepts within the same cluster have the same color

Percent increase or decrease, 89	Finding fractions and ratios, 74	Scientific notation, 67	Finding slope from situation, 111	Interior angles figures with more than 3 sides, 86
Scale factor, 94	Multiplication fractions, 40	Ordering positive decimals, 49	Distributive property, 84	Reflection, 150
Proportion, 32	Division fractions, 41	Ordering integers, 71	Solving inequalities, 105	Mean-median-mode-range differentiation, 146
Finding percents, 66	Exponents, 33	Order of operations +,-,/,* () positive reals, 83	Interior angles triangle, 23	D.4.8-understanding-concept-of-probabilities, 153
Simplifying expressions positive exponents, 38	Greatest common factor, 114	Percents, 108	Area trapezoid, 143	Mode, 93
Percent of, 15	Prime number, 101	Least common multiple, 112	Volume prism, 142	Table, 48
Order of operations all, 137	Multiplying monomials, 100	Subtraction whole numbers, 42	Volume sphere, 147	Sampling techniques, 92
Addition and subtraction fractions, 99	Addition whole numbers, 2	Unit conversion within a system, 68	Volume pyramid, 149	Scatter plot, 7
Equivalent fractions, 39	Conversion of fraction decimals percents, 60	Equation solving two or fewer steps, 123	Translations, 152	Estimation, 76
Division whole numbers, 65	Square roots, 122	Picking equation and inequality from Choices, 20	Calculations with similar figures, 26	Range, 63
Divisibility rules, 103	Multiplication whole numbers, 4	Finding y-intercept from linear equation, 18	Parallel and perpendicular slopes, 19	Median, 69

(b) Manually-labeled knowledge concept

Fig. 8 Clustering of embeddings of concepts and the concepts' labels. The number in (b) represents the index of a specific concept.

each other. While it is true that not all clustering results are accurate, the concept embeddings learned through the LTKT model still prove valuable in assisting educational experts in uncovering the interconnectedness among concepts more effectively.

6 Conclusion

In this study, a novel KT framework is introduced, which leverages deep learning techniques, namely positive and negative LTKT. By concurrently considering the positive and negative learning transfer effects on knowledge structure, LTKT better models the learning transfer relations among pedagogical concepts compared with recent works. Specifically, LTKT first proposes a statistical approach to build a learning transfer graph that models both the positive and negative learning transfer relations of concepts from a given course. A group of components is then carefully designed in LTKT to ensure the reasonableness of updating a student's knowledge states for all concepts, based on the learning transfer graph. Extensive experiments are conducted on two publicity physical world datasets, and the experimental results demonstrate the effectiveness and interpretability of the proposed method compared with baselines.

One limitation of LTKT is that it assumes that an exercise corresponds to only one concept, which is a typical setting in recent KT models[6, 12, 14, 16, 40]. In the future, we will consider the situation in which an exercise investigates multiple concepts. Another limitation of LTKT is that it relies on a statistical methodology to identify learning transfer relations among concepts. This approach may prove ineffective when dealing with concepts that are not well-represented in student learning behaviours, i.e., the cold start problem[46, 47]. In the future, we will utilize external knowledge to assist in generating learning transfer relations. Meanwhile, when modelling the learning transfer effect, LTKT only considers the learning interactions among concepts and fails to consider other factors (e.g. question-answering time and concept difficulty) that may influence the learning transfer effect. This is another area for improvement that we will address in the future. Finally, we would like to apply the proposed LTKT model to better address popular educational problems, such as peer assessment[48] and personalized learning materials recommendation[49].

Acknowledgment

This work was supported by the Guang Dong Basic and Applied Basic Research Foundation (No. 2025A1515011582), the National Natural Science Foundation of China (Nos. 62137001 and U20A20229), the Academician Fang Binxing Workstation in Hainan Province, China (No. YSGZZ2023003), and the Specific Research Fund of the Innovation Platform for Academicians of Hainan Province, China (No. YSPTZX202506).

References

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

[2] T. Liu, W. Chen, L. Chang, and T. Gu, Research advances in the knowledge tracing based on deep learning, (in Chinese), *J. Comput. Res. Dev.*, vol. 59, no. 1, pp. 81–104, 2022.

[3] S. Shen, E. Chen, Q. Liu, Z. Huang, W. Huang, Y. Yin, Y. Su, and S. Wang, Monitoring student progress for learning process-consistent knowledge tracing, *IEEE Trans. Knowledge Data Eng.*, vol. 35, no. 8, pp. 8213–8227, 2023.

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

[5] A. Ghosh, N. Heffernan, and A. S. Lan, Context-aware attentive knowledge tracing, in *Proc. 26th Int. ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 2330–2339.

[6] S. Shen, Q. Liu, E. Chen, Z. Huang, W. Huang, Y. Yin, Y. Su, and S. Wang, Learning process-consistent knowledge tracing, in *Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2021, pp. 1452–1460.

[7] T. Long, Y. Liu, J. Shen, W. Zhang, and Y. Yu, Tracing knowledge state with individual cognition and acquisition estimation, in *Proc. 44th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, 2021, pp. 173–182.

[8] A. T. Corbett and J. R. Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, *User Model. User-Adap. Inter.*, vol. 4, no. 4, pp. 253–278, 1994.

[9] H. Cen, K. Koedinger, and B. Junker, Learning factors analysis-a general method for cognitive model evaluation and improvement, in *Proc. 8th Int. Conf. Intelligent Tutoring Systems*, Jhongli, China, 2006, pp. 164–175.

[10] P. I. Pavlik, H. Cen, and K. R. Koedinger, Performance factors analysis--a new alternative to knowledge tracing, in *Proc. 2009 Conf. Artificial Intelligence in Education: Building Learning Systems That Care: From Knowledge Representation to Affective Modelling*, Brighton, UK,

2009, pp. 531–538.

[11] J. J. Vie and H. Kashima, Knowledge tracing machines: Factorization machines for knowledge tracing, in *Proc. 33rd Int. AAAI Conf. Artificial Intelligence*, Honolulu, HI, USA, 2019, pp. 750–757.

[12] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. Guibas, and J. Sohl-Dickstein, Deep knowledge tracing, in *Proc. 29th Int. Conf. Neural Information Processing Systems*, Montreal, Canada, 2015, pp. 505–513.

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

[14] S. Shen, Q. Liu, E. Chen, H. Wu, Z. Huang, W. Zhao, Y. Su, H. Ma, and S. Wang, Convolutional knowledge tracing: Modeling individualization in student learning process, in *Proc. 43rd Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Virtual Event, 2020, pp. 1857–1860.

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

[16] S. Tong, Q. Liu, W. Huang, Z. Hunag, E. Chen, C. Liu, H. Ma, and S. Wang, Structure-based knowledge tracing: An influence propagation view, in *Proc. 2020 IEEE Int. Conf. Data Mining*, Sorrento, Italy, 2020, pp. 541–550.

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

[18] E. C. Butterfield and G. D. Nelson, Theory and practice of teaching for transfer, *Educ. Technol. Res. Dev.*, vol. 37, no. 3, pp. 5–38, 1989.

[19] R. Burns and C. B. Dobson, Transfer of learning (training), in *Introductory Psychology*, R. B. Burns and C. B. Dobson, eds. Dordrecht, the Netherlands: Springer, 1984, pp. 345–357.

[20] D. N. Perkins and G. Salomon, Transfer of learning, in *The International Encyclopedia of Education*, T. Husén and T. N. Postlethwaite, eds, 2nd ed. Oxford, UK: Pergamon, 1992, pp. 6452–6457.

[21] P. G. Sidney and M. W. Alibali, Making connections in math: Activating a prior knowledge analogue matters for learning, *J. Cogn. Dev.*, vol. 16, no. 1, pp. 160–185, 2015.

[22] S. Minn, J. J. Vie, K. Takeuchi, H. Kashima, and F. Zhu, Interpretable knowledge tracing: Simple and efficient student modeling with causal relations, in *Proc. 36th Int. AAAI Conf. Artificial Intelligence*, Virtual Event, 2022, pp. 12810–12818.

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

[24] C. K. Yeung and D. Y. Yeung, Addressing two problems in deep knowledge tracing via prediction-consistent regularization, in *Proc. 5th Annu. ACM Conf. Learning at Scale*, London, UK, 2018, p. 5.

[25] W. Zhao, J. Xia, X. Jiang, and T. He, A novel framework for deep knowledge tracing via gating-controlled forgetting and learning mechanisms, *Inf. Process. Manage.*, vol. 60, no. 1, p. 103114, 2023.

[26] Z. Liu, Q. Liu, J. Chen, S. Huang, B. Gao, W. Luo and J. Weng, Enhancing deep knowledge tracing with auxiliary tasks, in *Proc. ACM Web Conf. 2023*, Austin, TX, USA, 2023, pp. 4178–4187.

[27] Y. Zhao, H. Ma, W. Wang, W. Gao, F. Yang, and X. He, Exploiting multiple question factors for knowledge tracing, *Expert Syst. Appl.*, vol. 223, p. 119786, 2023.

[28] S. Zhao and S. Sahebi, Graph-enhanced multi-activity knowledge tracing, in *Proc. 34th European Conf. Machine Learning and Knowledge Discovery in Databases*: *Applied Data Science and Demo Track*, Turin, Italy, 2023, pp. 529–546.

[29] Y. Kou, X. Zhang, Y. Yang, Y. Hu, B. Wu, and J. Liu, A learner knowledge state modeling method based on the improved DKVMN, in *Proc. 2023 IEEE Smart World Congress*, Portsmouth, UK, 2023, pp. 1–4.

[30] E. Tsutsumi, Y. Guo, R. Kinoshita, and M. Ueno, Deep knowledge tracing incorporating a hypernetwork with independent student and item networks, *IEEE Trans. Learn. Technol.*, vol. 17, pp. 951–965, 2024.

[31] W. Zhang, Z. Gong, P. Luo, and Z. Li, DKVMN-KAPS: Dynamic key-value memory networks knowledge tracing with students' knowledge-absorption ability and problem-solving ability, *IEEE Access*, vol. 12, pp. 55146–55156, 2024.

[32] W. Gan, Y. Sun, and Y. Sun, Knowledge structure enhanced graph representation learning model for attentive knowledge tracing, *Int. J. Intell. Syst.*, vol. 37, no. 3, pp. 2012–2045, 2022.

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

[34] Z. Wang, X. Feng, J. Tang, G. Y. Huang, and Z. Liu, Deep knowledge tracing with side information, in *Proc. 20th Int. Conf. Artificial Intelligence in Education*, Chicago, IL, USA, 2019, pp. 303–308.

[35] M. J. Nathan and M. W. Alibali, An embodied theory of transfer of mathematical learning, in *Transfer of Learning*, C. Hohensee and J. Lobato, eds. Cham, Switzerland: Springer, 2021, pp. 27–58.

[36] H. Liu, T. Zhang, F. Li, Y. Gu, and G. Yu, Tracking knowledge structures and proficiencies of students with learning transfer, *IEEE Access*, vol. 9, pp. 55413–55421, 2021.

[37] M. Yu, F. Li, H. Liu, T. Zhang, and G. Yu, ContextKT: A context-based method for knowledge tracing, *Appl. Sci.*, vol. 12, no. 17, p. 8822, 2022.

[38] S. Zu, L. Li, and J. Shen, CAKT: Coupling contrastive learning with attention networks for interpretable knowledge tracing, in *Proc. 2023 Int. Joint Conf. Neural Networks*, Gold Coast, Australia, 2023, pp. 1–8.

[39] S. Huang, Z. Liu, X. Zhao, W. Luo, and J. Weng, Towards robust knowledge tracing models via k-sparse attention, in *Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, Taipei, China, 2023, pp. 2441–2445.

[40] M. Chen, Q. Guan, Y. He, Z. He, L. Fang, and W. Luo,

Knowledge tracing model with learning and forgetting behavior, in *Proc. 31st ACM Int. Conf. Information & Knowledge Management*, Atlanta, GA, USA, 2022, pp. 3863–3867.

[41] X. Xiong, S. Zhao, E. Van Inwegen, and J. E. Beck, Going deeper with deep knowledge tracing, in *Proc. 9th Int. Conf. Educational Data Mining*, Raleigh, NC, USA, 2016, pp. 545–550.

[42] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder–decoder for statistical machine translation, in *Proc. 2014 Conf. Empirical Methods in Natural Language Processing*, Doha, Qatar, 2014, pp. 1724–1734.

[43] X. Glorot and Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proc. 13th Int. Conf. Artificial Intelligence and Statistics*, Sardinia, Italy, 2010, pp. 249–256.

[44] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv: 1412.6980, 2015.

[45] L. van der Maaten and G. Hinton, Visualizing data using t-SNE, *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[46] J. Pang, Y. Huang, Z. Xie, J. Li, and Z. Cai, Collaborative city digital twin for the COVID-19 pandemic: A federated learning solution, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 759–771, 2021.

[47] Y. Peng, S. Xu, Q. Chen, W. Huang, and Y. Huang, A novel popularity extraction method applied in session-based recommendation, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 971–984, 2024.

[48] Q. Sun, J. Wu, W. Rong, and W. Liu, Formative assessment of programming language learning based on peer code review: Implementation and experience report, *Tsinghua Science and Technology*, vol. 24, no. 4, pp. 423–434, 2019.

[49] S. Dai, Y. Yu, H. Fan, and J. Dong, Spatio-temporal representation learning with social tie for personalized POI recommendation, *Data Sci. Eng.*, vol. 7, pp. 44–56, 2022.

Jia Xu received the BEng and MEng degrees in software engineering from Northeastern University, China in 2006 and 2008, respectively, and the PhD degree in computer science from Northeastern University, China in 2013. She is a professor at Cyberspace Institute of Advanced Technology, Guangzhou University, China. Her current research interests include knowledge tracing, personalized recommendation, and data management. She is a distinguished member and a director of China Computer Federation (CCF).

Pin Lv received BEng degree in software engineering from Northeastern University, China in 2006, and the PhD degree in computer science and technology from National University of Defense Technology (NUDT), China in 2012. He is a professor at Cyberspace Institute of Advanced Technology, Guangzhou University, China. His research interests include artificial intelligence, educational data analysis, and Internet of things. He is a distinguished member of CCF.

Ge Yu received the BEng and MEng degrees in computer science from Northeastern University, China in 1982 and 1986, respectively, and the PhD degree in computer science from Kyushu University, Japan in 1996. He is a professor at School of Computer Science and Engineering, Northeastern University, China. His current research interests include distributed and parallel systems, cloud computing and big data management, and blockchain techniques and systems. He is a fellow of CCF, a senior member of Association for Computing Machinery (ACM), and a senior member of IEEE.

Rongrong Tang received the BEng degree in computer science and technology from Guangxi University, China in 2019. She is a master student at School of Computer, Electronics and Information, Guangxi University. Her research interests include knowledge tracing, educational data mining, and intelligent tutoring system.

Minghe Yu received the BEng degree in computer science and technology from Northeastern University, China in 2012, and the PhD degree in computer science and technology from Tsinghua University, China in 2018. She is an associate professor at Software College, Northeastern University, China. Her research interests include big data, information retrieval, and data mining.

Enhong Chen received the BEng degree in computer science from Anhui University, China in 1989, the MEng degree in computer science from Hefei University of Technology, China in 1992, and the PhD degree in computer science from University of Science and Technology of China (USTC), China in 1996. He is a professor at School of Computer Science, USTC, China, and a vice dean at Faculty of Information and Intelligence of USTC, China. His research interests include data mining and machine learning, social network analysis, and recommender systems. He is a fellow of CCF, and a fellow of IEEE.