Structure-based Knowledge Tracing: An Influence Propagation View

Shiwei Tong[1], Qi Liu[1,*], Wei Huang[1], Zhenya Huang[1], Enhong Chen[1], Chuanren Liu[2], Haiping Ma[3], Shijin Wang[4,5]
[1]Anhui Province Key Laboratory of Big Data Analysis and Application, School of Data Science &
School of Computer Science and Technology, University of Science and Technology of China
[2]Business Analytics and Statistics, University of Tennessee, [3]Anhui University,
[4]iFLYTEK Research, iFLYTEK CO., LTD., [5]State Key Laboratory of Cognitive Intelligence,
{tongsw, ustc0411}@mail.ustc.edu.cn, {qiliuql,huangzhy, cheneh}@ustc.edu.cn
chuanren@xminer.org, hpma2020@163.com, sjwang3@iflytek.com

***Abstract*—In online education, Knowledge Tracing (KT) is a fundamental but challenging task that traces learners' evolving knowledge states. Much attention has been drawn to this area and several works such as Bayesian Knowledge Tracing and Deep Knowledge Tracing are proposed. Recent works have explored the value of relations among concepts and proposed to introduce knowledge structure into KT task. However, the propagated influence among concepts, which has been shown to be a key factor in human learning by the educational theories, is still under-explored. In this paper, we propose a new framework called Structure-based Knowledge Tracing (SKT), which exploits the multiple relations in knowledge structure to model the influence propagation among concepts. In the SKT framework, we not only consider the temporal effect on the exercising sequence but also take the spatial effect on the knowledge structure into account. We take advantages of two novel formulations in modeling the influence propagation on the knowledge structure with multiple relations. For undirected relations such as similarity relations, the synchronization propagation method is adopted, where the influence propagates bidirectionally between neighbor concepts. For directed relations such as prerequisite relations, the partial propagation method is applied, where the influence can only unidirectionally propagate from a predecessor to a successor. Meanwhile, we employ the gated functions to update the states of concepts temporally and spatially. Extensive experiments demonstrate the effectiveness and interpretability of SKT.**

***Index Terms*—Transfer of knowledge; Knowledge Tracing; Influence Propagation; Recurrent Neural Network;**

I. Introduction

Recent years have witnessed the booming of online education systems, such as *KhanAcademy.org* and *Junyiacademy.org*. These systems can not only assist tutors to give proper instruction based on the individual characteristics, e.g., strengths and weaknesses, of learners, but also help learners be aware of their learning progress. The conveniences and rapid developments have attracted increasing attention of educators and public [13], [19]. A key issue in the online education systems is Knowledge Tracing, the goal of which is to precisely trace the evolving knowledge states of learners on the concepts based on their past exercising performance.

Traditional Knowledge Tracing models [7], [25], [41] mainly leverage the temporal information (i.e., learners' sequential performance on the exercises). For example, Bayesian Knowledge Tracing (BKT) [7] employs a hidden markov model to respectively trace the evolving knowledge state of each concept while Deep Knowledge Tracing (DKT) [25] uses the recurrent neural networks to jointly model the states of all concepts. Recently, more and more works [23], [34], [35] have noticed the value of the knowledge structure, which contains abundant domain knowledge. Chen et al. [4] used the prerequisite relations in knowledge structure to reformulate knowledge tracing as a constraint problem and Nakagawa et al. [23] utilized graph neural networks on a homogeneous graph knowledge structure to enhance knowledge tracing. Although with significant improvement by utilizing knowledge structure, previous works ignore the propagated influence among concepts.

According to one of education theories, *transfer of knowledge* [8], [31], [36], not only the proficiency of the current learning concept but also some relevant concepts will be changed when a learner learns a concept. As illustrated in the middle part of Figure 1, a learner practices several exercises on concepts B, D, ..., C, D sequentially and correctness (right or wrong) of the answer given by learner is shown under the concepts. The concept and correctness of the answer at each time step are called an exercise-performance pair. The bottom part shows the knowledge structure. The vertexes are the pedagogical concepts and are linked by multiple relations. The multiple relations include not only directed relations but also undirected relations. Without loss of generality, here we use two typical relations as a toy example. In Figure 1, the black directed lines represent prerequisite relations[1] and blue undirected lines stand for similarity relations[2]. At the most beginning, after the learner finishes the learning on concept B, her proficiency on concept B increases, which can be seen from the radar graphs on the top of Figure 1. Meanwhile, the proficiency of concepts linked by multiple relations is

[1]A concept points to another concept with prerequisite relation means the former one is considered as the foundation of the later one.

[2]Concepts linked by similarity relations is somehow similar in the content.

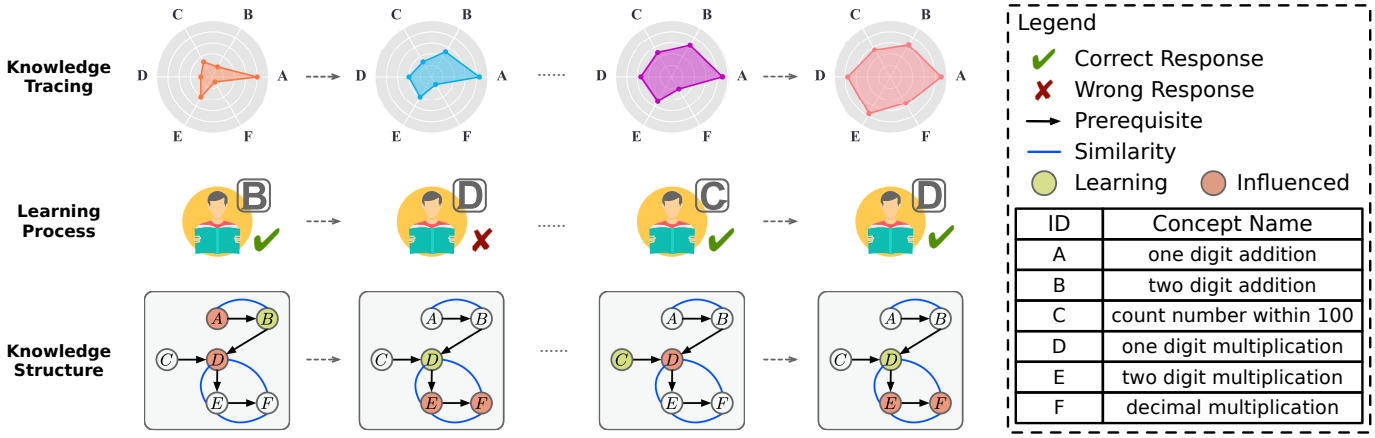


Fig. 1: Illustration of Knowledge Tracing. $B \rightarrow D \rightarrow ... \rightarrow C \rightarrow D$ is an exercises sequence where each exercise corresponds to one concept. The knowledge structures showed in the bottom contain two types of relations (i.e., prerequisite and similarity). Radar graphs in the top show the evolving proficiency on each concepts during learning. Graphs in the bottom indicate the knowledge structure where current-learning concepts are highlighted in green and influenced concepts in red.

also influenced. For instance, the proficiency on concept D (a successor of B) and concept A (a concept similar with B) also increase. The reason why the learning on concept C influences the proficiency on concept A, D is that the knowledge can be transferred among concepts. In other words, the learning influence can be propagated along the multiple relations in the knowledge structure. Thus, it is essential to consider the influence propagation when utilizing the knowledge structure for knowledge tracing.

However, there are two major challenges along this line. First, the knowledge state of each concept is determined by two types of effects. One is the temporal effect from the exercise sequence and the other one is the spatial effect from the knowledge structure. As shown in Figure 1, at each time step, when a learner practices an exercise, the learning behaviour results in a temporal effect on the learning concept, which changes the state of the concept (e.g., the state on concept B is changed by the learning at the first step); Then, the variation of states of the learning concept will furthermore influence its neighbors and successors in the knowledge structure through different relations, which is called spatial effect. Thus, there are two dimensions of learning effects which we need to simultaneously model. How to jointly model the temporal and spatial effect is a challenging problem. Second, it is not easy to model the spatial effect on a knowledge structure with multiple relations. Because the influence can be propagated along different relations, a key issue is to consider the different influence propagation ways on different types of relations. As shown in Figure 1, there are multiple relations in the knowledge structure, including directed relations and undirected relations. Therefore, when we model the spatial effect, the influence propagation on different types of relations needs to be respectively considered.

To address the challenges above, we propose a new framework called Structure-based Knowledge Tracing (SKT), which can concurrently model the temporal and spatial effects. Specifically, at each time step, we first extract the temporal effect from the exercise-performance pair and update the state of the practiced concept via a gated function. Then, to model the influence propagation in the knowledge structure caused by the temporal effect, we apply the synchronization and partial propagation methods to characterize the undirected and directed relations among knowledge structure, respectively. Finally, for those influenced concepts, the same gated function as mentioned above will be used to update the states based on the influence propagated to them. In this way, we model the influence propagation in the knowledge structure and furthermore jointly model the temporal and spatial effect. Extensive experiments on real-world datasets show that SKT not only significantly outperforms several baselines, but also effectively provides interpretable insights for understanding the evolving states of learners.

II. Related Work

Generally, the related works of this study are grouped into the following two categories.

A. Knowledge Tracing

Knowledge tracing is a task of modeling learners' knowledge states over time so that we are able to accurately predict how learners will perform on future exercises [13]. One of the classical knowledge tracing models is Bayesian Knowledge Tracing (BKT) [7]. BKT-based approach models learner's knowledge in a Hidden Markov Model (HMM) as a set of binary variables, which represents whether the learner has mastered a skill or not (e.g., 0 indicates no while 1 indicates mastered). As deep learning models outperform the conventional models in a range of domains such as pattern recognition and natural language processing, Piech et al. [25] used RNN to model the evolving proficiency on concepts and proposes the Deep Knowledge Tracing (DKT) model. Different from BKT using the binary variables to represent

the learner's knowledge states, by using Recurrent Neural Network (RNN), DKT models such states in a high-dimensional and continuous representation. Another kind of deep learning models is Deep Key-Value Memory Networks (DKVMN) [41]. DKVMN facilitates one static key memory matrix and one dynamic value memory matrix. The key memory matrix stores the knowledge concepts and the value memory matrix stores and updates the mastery levels of corresponding concepts. DKVMN is able to automatically learn the correlation between input exercises and underlying concepts. DKT and DKVMN encourage increasing amounts of research on deep learning-based knowledge tracing models [22], [39].

Recently, more and more works have paid attention to introduce the knowledge structure into knowledge tracing. Chen et al. [4] and Wang et al. [35] respectively proposed a regularization term based on the prerequisite and similarity relations. Wang et al. [34] used the hierarchical knowledge structure and put forward the Deep Hierarchical Knowledge Tracing (DHKT) model while Nakagawa et al. [23] introduced the Graph Neural Network (GNN) into knowledge tracing with a graph-like knowledge structure. Nevertheless, previous works ignore the influence among concepts during learning or can only handle the knowledge structure with one-type relations, which somehow limits their performance.

B. Influence Propagation

Several models [14], [15], [33] have been provided to describe the dynamics of influence propagation. These models define the stochastic process of information propagation. Thus they are called stochastic diffusion models. Among them, the Independent Cascade model (IC) and Linear Threshold model (LT) have been widely used and studied [14], [20]. In both models, the influence spread is simply defined as the expected number of activated nodes. Recently, some authors proposed to introduce neural networks to influence propagation models [1], [17], [37]. Atwood et al. [1] presented diffusion-convolutional neural networks to learn diffusion-based representations from graph-structured data and used as an effective basis for node classification. Li et al. [17] proposed Diffusion Convolutional Recurrent Neural Network (DCRNN) on traffic forecasting to incorporate both spatial and temporal dependency in the traffic flow. These methods receive a graph with a single relation type, which makes it hard to be directly applied in our task.

III. Problem Formulation

Before formally introducing SKT, we give the necessary definitions as follows:

A. Knowledge Structure

Educational theories have emphasized the importance of knowledge structure [24], [26], which contains many relations such as prerequisite [4], [28] and similarity [35]. Prerequisite indicates the hierarchical structure existing among the learning items. As represented in bottom graphs of Figure 1, the directed arrow from one vertex to the other means that the former is a prerequisite for the latter, e.g., *count number within 100* is a prerequisite for *one digit multiplication*. Similarity is another widely studied relation. As illustrated in Figure 1, the vertexes linked by the blue undirected edge (i.e., similarity) are involved in the same topic or area and may overlap in some knowledge.

Definition 1: (Knowledge Structure) In this paper, the knowledge structure with multiple relations is represented as a graph $G(V, E)$, where $V = \{v_1, v_2, ..., v_N\}$ and each vertex v corresponds to one concept. There are multiple relations $E = \{E^r, r = 1, ..., R\}$, where r stands for a certain type of relations (e.g., prerequisite and similarity) and E^r represents all relations of the type r. R is the number of relation types.

B. Problem Statement

Knowledge tracing task consists of two parts: (1) modeling a learner's knowledge state through their performance sequence and (2) predicting how a learner will perform on future exercises. Knowledge tracing task is usually formulated as a supervised sequence prediction problem. By introducing the graph-like knowledge structure G into knowledge tracing problem, we formulate this knowledge tracing problem as:

Definition 2: (Knowledge Tracing with Knowledge Structure) Given a learner's past exercise sequence of exercise-performance pairs, i.e., $X = \{x_t, t = 1, ..., T\}$, where $x_t = (e_t, p_t)$. $p_t \in \{0, 1\}$ is the correctness (i.e., 0 indicates the learner giving a wrong answer while 1 indicates giving a correct one.) of the learner's answer on the exercise e_t at the step t. Each exercise e_t tests one concept c_t. Each concept corresponds to one vertex v in the knowledge structure $G(V, E)$. Our goal is to model the learner's knowledge states $\mathcal{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T\}$ on all N concepts (i.e., the vertexes V in G), and predict the probability that the learner will correctly answer a new exercise e_{t+1} when given the learner's past exercise sequence $x_{1,...,t}$ and the knowledge structure G, i.e., $P(p_{t+1} = 1 | e_{t+1}, x_{1,...,t}, G)$.

IV. Structure-based Knowledge Tracing

This section begins with a brief overview of our framework. The components of SKT are then introduced in detail.

A. Overview

SKT is a sequential model, which leverages the graph-structured nature of knowledge and applies two different propagation models to trace the influence along different relations. We present the architecture of SKT in Figure 2. At each time step t, a d_h-dimension vector $\boldsymbol{h}_i^t$ is used to represent the hidden state on concept i. The learner's hidden state vectors on all concepts form up the hidden states $\mathcal{H}$, as shown in the left-top part of Figure 2. A Cascade Influence Propagation (CIP) unit is used to jointly model the temporal and spatial effects on concepts. At each time step t, the CIP unit first extracts the temporal effect on the current practice concept from the exercise-performance pair $x_t = (e_t, p_t)$. After that, some other concepts are spatially affected after the temporal effect on concept i. To model the spatial effect on different types of relations, we propose two different propagation methods: partial

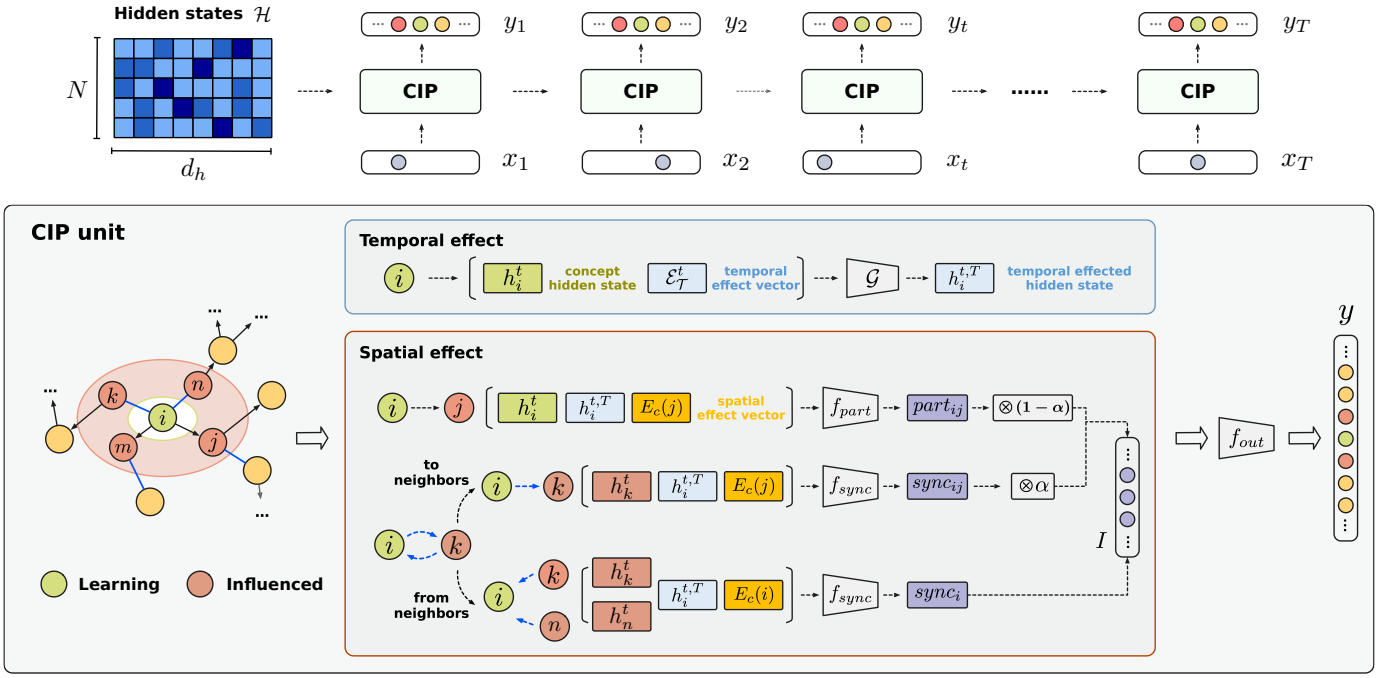


Fig. 2: The overview of Structure-based Knowledge Tracing (SKT).

propagation method for directed relations and synchronization propagation method for undirected relations. A gated function is then adopted to update the hidden state based on temporal and spatial effect. To predict whether the learner will correctly answer a new exercise, a map function $f_{out}(\boldsymbol{h}_i^t)$ is used to infer the correctly answering probability based on the hidden state on concept i. The following paragraphs explain the processes in detail.

B. Modelling of Temporal Effect

Based on the educational studies on concept learning [11] and previous works in KT [7], [25], [41], when a learner practices the exercise, a learning effect will be generated and acts on the learning concept. As shown in Figure 2, at each time step t, a temporal learning effect $\boldsymbol{\mathcal{E}}_{\mathcal{T}}^t$ acts on concept i, which changes the hidden state on concept i from $\boldsymbol{h}_i^t$ to $\boldsymbol{h}_i^{t,T}$. The temporal effect on current learning concept is implied based on the exercise-performance pair $x_t = \{e_t, p_t\}$, where e_t tests the concept i. Similar to previous works [25], [38], [40], a performance vector $\boldsymbol{x}^t \in \{0,1\}^{2N}$ is used to represent the exercise-performance pair x_t:

$$\boldsymbol{x}_j^t = \begin{cases} 1 & \text{if } j = 2 \cdot e_t + p_t, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then we embed the performance vector to formulate the temporal effect vector $\boldsymbol{\mathcal{E}}_{\mathcal{T}}^t$:

$$\boldsymbol{\mathcal{E}}_{\mathcal{T}}^t = \boldsymbol{x}^t \mathbf{E}_r, \tag{2}$$

where $\boldsymbol{E}_r \in \mathbb{R}^{2N \times d_e}$ is a matrix embedding the performance vector $\boldsymbol{x}^t$. The temporal effect vector $\boldsymbol{\mathcal{E}}_{\mathcal{T}}^t$ is then input into a gated function $\mathcal{G}$ to update the state of the concept i:

$$\boldsymbol{h}_i^{t,T} = \mathcal{G}(\boldsymbol{\mathcal{E}}_{\mathcal{T}}^t, \boldsymbol{h}_i^t), \tag{3}$$

where $\mathcal{G}(\bullet, \bullet)$ is the Gated Recurrent Unit (GRU) gate [5][3]. Next, the follow-up parts will elaborate on how other concepts are spatially affected after the temporal effect on concept i.

C. Modelling of Spatial Effect

Once the state of concept i is changed, the influence will be propagated to the related concepts along the multiple relations. As illustrated in the left-bottom part of Figure 2, the hidden state of concept j is changed by the propagated influences from concept i. The following parts will thoroughly describe the two different influence propagation methods: partial propagation and synchronization propagation.

1) Partial Propagation: For those directed relations, such as prerequisite relations [4] and remedial relations [28], we adopt a partial propagation method. Among the direct relations, prerequisite relations is the most well studied one. Previous works [4], [23] have established the ordering relation of the proficiency of predecessor concepts and successor concepts, where the proficiency of the former one is expected to be higher than the latter. The conclusion can be further explained from the perspective of transfer of knowledge [29]: the influence is unidirectionally propagated from only predecessors to successors. Therefore, we propose the partial propagation method, which generates the influence based on the variation of the state of predecessor concepts and propagates the influence to successor concepts along directed relations. Concretely, as shown in Figure 2, after the temporal effect where the hidden state on concept i is changed from $\boldsymbol{h}_i^t$ to

[3]Here we use GRU for it is computationally more efficient with less complex structure compared with Long Short Term Memory (LSTM) [10], [18]. However it should be noticed that the performance of LSTM is on a par with GRU [6], which means GRU can also be replaced by LSTM.

$\boldsymbol{h}_i^{t,T}$, the variation of the state on concept i will result in an influence and be propagated along the directed relations to its successors:

$$
\begin{aligned}
&part_{ij}^r = f_{part}(\boldsymbol{h}_i^{t,T}, \boldsymbol{h}_i^t, \boldsymbol{E}_c(j)), \forall j \in \mathcal{S}^r(i), \\
&f_{part}(\boldsymbol{h}_i^{t,T}, \boldsymbol{h}_i^t, \boldsymbol{E}_c(j)) = relu(\mathbf{W}_p^r \boldsymbol{P}_{ij}^r + \mathbf{b}_p^r), \\
&\boldsymbol{P}_{ij}^r = (\boldsymbol{h}_i^{t,T} - \boldsymbol{h}_i^t) \oplus \boldsymbol{E}_c(j).
\end{aligned} \tag{4}
$$

$\mathcal{S}^r(i)$ is a successorhood function, which returns all successor concepts of i on r. $\mathbf{W}_p^r$ and $\boldsymbol{b}_p^r$ are learned parameters. $\oplus$ is the operation that concatenates two vectors into a long vector. In addition to the variation of the state on concept i, we also include a vector $\boldsymbol{E}_c(j)$ to represent the concept feature. $\boldsymbol{E}_c \in \mathbb{R}^{N \times d_c}$ is a matrix embedding the concept index, where N is the number of concepts and d_c is the embedding size, and $\boldsymbol{E}_c(j)$ represents the j-th row of $\boldsymbol{E}_c$.

2) Synchronization Propagation: Previous works on undirected relations, such as similarity relations [35] and collaboration relations [12], have got some interesting conclusions. Wang et al. [35] found that, in similarity relations, the promotion of the proficiency of a certain concept brought the promotion to its neighbor concepts and vice versa, which results in the similar proficiency of the neighbor concepts. The idea can be further explained based on the theories of transfer of knowledge [27], the influence is bidirectionally propagated between neighbor concepts. Inspired by these observations, we propose a synchronization propagation method to model the bidirectional influence propagation. Similar to partial propagation, after the temporal effect where the hidden state on concept i is changed from $\boldsymbol{h}_i^t$ to $\boldsymbol{h}_i^{t,T}$, the variation of the state on i will be result in an influence and be propagated along the undirected relations to its successors. To be noticed that, different from partial propagation which is unidirectional where the propagated influence is only decided by the variation of concept i, in synchronization propagation, the influence is determined by the state on both i and its neighbors and will be propagated bidirectionally. Specifically, we use two formulations to respectively model the influence propagated from the concept i to its neighbors and the influence propagated from the neighbors to the concept i.

We first use the following formulation to model the influence propagated from the concept i to its neighbors:

$$
\begin{aligned}
&sync_{ij}^r = f_{sync}(\boldsymbol{h}_i^{t,T}, \boldsymbol{h}_j^t, \boldsymbol{E}_c(j)), \forall j \in \mathcal{N}^r(i), \\
&f_{sync}(\boldsymbol{h}_i^{t,T}, \boldsymbol{h}_j^t, \boldsymbol{E}_c(j)) = relu(\mathbf{W}_s^r \boldsymbol{S}_{ij}^r + \boldsymbol{b}_s^r), \\
&\boldsymbol{S}_{ij}^r = \boldsymbol{h}_i^{t,T} \oplus \boldsymbol{h}_j^t \oplus \boldsymbol{E}_c(j).
\end{aligned} \tag{5}
$$

$\mathcal{N}^r(i)$ is a neighborhood function, which returns all neighbor concepts of i on r. $\mathbf{W}_s^r$ and $\boldsymbol{b}_s^r$ are learned parameters. $\boldsymbol{E}_c$ is the same embedding matrix as Section IV-C1 and $\boldsymbol{E}_c(j)$ represents the concept feature.

Then, we model the influence propagated from the neighbors of concept i to itself:

$$
\begin{aligned}
&sync_i^r = relu(\mathbf{W}_{ss}^r \boldsymbol{R}_i^r + \boldsymbol{b}_{ss}^r), \\
&\boldsymbol{R}_i^r = (\boldsymbol{h}_i^{t,T} + \sum_{j \in \mathcal{N}^r(i)} \boldsymbol{h}_j^t) \oplus \boldsymbol{E}_c(i),
\end{aligned} \tag{6}
$$

Algorithm 1 Cascade Influence Propagation.

Input: Knowledge Structure G(V, E); performance vector $\boldsymbol{x}^t$; current learning concept i; neighborhood function $\mathcal{N}^r$ for a specific relation type r; successorhood function $\mathcal{S}^r$ for a specific relation type r; current state of concepts $\mathcal{H}^t = \{\boldsymbol{h}_v^t, \forall v \in V\}$; The relations to apply synchronization propagation method $R^S = \{r_1^S, r_2^S ...\}$; The relations to apply partial propagation method $R^P = \{r_1^P, r_2^P, ..., \}$;.

Output: States of all concepts at next step $\mathcal{H}^{t+1} = \{\boldsymbol{h}_v^{t+1}, \forall v \in V\}$;

```
h_i^{t,T} = G(x^t E_r, h_i^t) // Temporal effect (Section IV-B)
for r in R_P do // Partial propagation (Section IV-C1)
    for j in S^r(i) do
        part_ij^r = f_part(h_i^{t,T}, h_i^t, E_c(j))
    end for
end for
for r in R_S do // Synchronization propagation (Section IV-C2)
    for j in N^r(i) do
        sync_ij^r = f_sync(h_i^{t,T}, h_j^t, E_c(j)),
        // from i to its neighbor j (Equation 5)
    end for
    sync_i^r = relu(W_ss^r((h_i^{t,T} + Σ_{j∈N^r(i)} h_j^t) ⊕ E_c(i)) + b_ss^r)
    // from neighbors to i (Equation 6)
end for
// update the hidden state (Section IV-D)
use Equation 7 and 8 to get I_j for each influenced concept j.
// update the hidden state on j (Equation 9)
h_j^{t+1} = G(I_j, h_j^t)
return H^{t+1}
```

where $\mathcal{N}^r(i)$ is the same neighborhood function as Equation 5. $\mathbf{W}_{ss}^r$ and $\boldsymbol{b}_{ss}^r$ are learned parameters.

In summary, synchronization propagation differs from partial propagation in two aspects: (1) the influence is only decided by the variation of concept i in partial propagation, while it is determined by the state on both i and its neighbors in synchronization propagation; (2) not only the neighbors but also the concept i are influenced during synchronization propagation, while only successors are influenced in partial propagation. These differences make synchronization propagation bidirectional and partial propagation unidirectional.

D. Update of Knowledge State

Next, for those concepts influenced by synchronization propagation or partial propagation, the model first aggregates the influences from both synchronization propagation and partial partial propagation and then updates the hidden states based on the aggregated influences. For each influenced concept j, the the aggregated influence I_j is calculated as:

$$
\boldsymbol{A}_j = \begin{cases} \sum_r sync_j^r & j = i, \\ \alpha \cdot \sum_r sync_{ij}^r + (1-\alpha) \cdot \sum_r part_{ij}^r & j \neq i, \end{cases} \tag{7}
$$

$$
\boldsymbol{I}_j = relu(\mathbf{W}_I \boldsymbol{A}_j + \mathbf{b}_I), \tag{8}
$$

where $\mathbf{W}_I$, $\mathbf{b}_I$ are learned weight matrix and bias, and α is a hyper-parameter. Then we use the following formulation to update the state on each influenced concept j:

$$
\boldsymbol{h}_j^{t+1} = \mathcal{G}(\boldsymbol{I}_j, \mathbf{h}_j^t), \tag{9}
$$

where $\mathcal{G}(\bullet, \bullet)$ is a GRU gate. The full process of influence propagation is shown in Algorithm 1.

E. Final Prediction

Finally, for each concept i, the model will output the predictive probability of a learner correctly answering the corresponding exercise at the next time step t:

$$\begin{aligned} \hat{p}_i^t &= f_{out}(\boldsymbol{h}_i^t), \\ f_{out}(\boldsymbol{h}_i^t) &= \sigma(\mathbf{W}_o \boldsymbol{h}_i^t + \mathbf{b}_o), \end{aligned} \tag{10}$$

where $\mathbf{W}_o$ is a learned weight matrix and $\mathbf{b}_o$ is a learned bias item. At time step t, the learner's knowledge state is calculated as: $\boldsymbol{y}_t = \{\hat{p}_1^t, ..., \hat{p}_N^t\}$.

The probability that the learner will correctly answer a new exercise e_t:

$$P(p_t = 1 | e_t, x_{1,...,t-1}, G) = \hat{p}_{e_t}^t. \tag{11}$$

F. Loss Function and Model Training

During the training stage, the parameters of SKT are jointly learned by minimizing a standard cross entropy loss between $\hat{p}_t$ and the true label p_t:

$$\mathcal{L} = -\sum_t (p_t log \hat{p}_t + (1 - p_t) log(1 - \hat{p}_t)). \tag{12}$$

SKT is fully differentiable and can be trained efficiently with stochastic gradient descent. The framework setting and training details are presented respectively in Section V-B2 and Section V-B3.

V. Experiments

In this section, we first introduce the datasets. Then, performance of SKT is compared with several baselines. At last, we show the interpretability of SKT.

A. Dataset

We use two real-world datasets, ASSISTments2014-2015 "skill-builder" dataset provided by the online educational service ASSISTments[4] and Junyi academy[5] [3] crawled from a Chinese e-learning platform. We preprocess each dataset using certain conditions and the preprocessed datasets are depicted in Table I, where ASSISTments2014-2015 is abbreviated as ASSISTments and Junyi academy is abbreviated as Junyi.

1) Junyi: The Junyi academy dataset includes a knowledge structure labeled by experts and learners' exercise performance logs in mathematics, where a learner has several exercise sequences. Each exercise-performance pair recorded in the learners' log contains the information of a learner for one exercise. Here is an example of one exercise performance sequence: {(*representing numbers*, correct), (*division 4*, wrong), (*conditional statements 2*, wrong), (*conditional statements 2*, wrong)}). Similar to [39], we select 1,000 most active learners from the exercise log to yield the dataset.

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

[5]https://pslcdatashop.web.cmu.edu/DatasetInfo?datasetId=1198

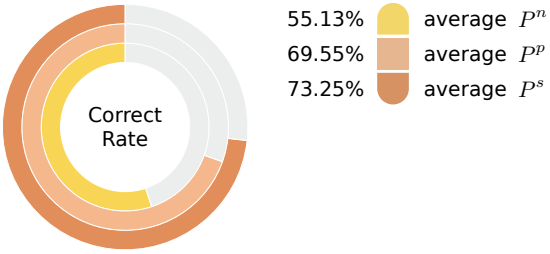


Fig. 3: Correct rate comparison.

The knowledge structure in Junyi academy contains two types of relations: the prerequisite relation and the similarity relation. They both contain several edges, e.g., in prerequisite relation, (*one digit addition*, *two digit addition*) stands for the linkage between the vertex *one digit addition* and the vertex *two digit addition* where the former is the prerequisite of the latter. In the prerequisite relation, we delete some loops in order to keep the graph to be a Directed Acyclic Graph (DAG). Due to the original data format of similarity relation is like ($concept_1$, $concept_2$, $similarity\_value$) (e.g., (*writing expressions 1*, *evaluating expressions 1*, 6.333)), where the $1 \leq similarity\_value \leq 9$. We set the threshold as 5.0 to get the similarity edges, i.e., $concept_1$ and $concept_2$ have an edge of similarity if $similarity\_value \geq 5.0$.

Furthermore, we investigate the practicing sequences of learners to verify the existence of the learning influence among concepts. Inspired by Piech et al. [25] and Nakagawa et al. [23], we use the following equation to calculate the correctness probability for concept pairs (i, j):$P_{ij} = \frac{n_c(j|i)}{n(j|i)}$, where $n_c(j|i)$ is the times that concept j is correctly answered in the first time step when its neighbor or predecessor i has been correctly answered, where $n(j|i)$ is the times that concept j is correctly answered. We respectively calculate the influence factor for prerequisite and similarity and denote them as P_{ij}^p and P_{ij}^s. As shown in Figure 3, compared with the non-conditional correctness probability $P_j^n = \frac{n_c(j)}{n(j)}$, we can see that when the neighbors and predecessors have been learned, the correctness probability of answering concept j is promoted. From the observation, we can conclude that there is some influence propagated from one concept to its neighbors or successors.

2) ASSISTments: We use the preprocessed dataset provided by Zhang et al. [41][6]. As the knowledge graph structure is not explicitly provided in the dataset, inspired by previous works [23], [25], we provide an implementation of constructing the graph structure.

Correct graph is a counting matrix, where $C_{ij} = c_{ij}$ if $i \neq j$; else, it is 0. Here, c_{ij} represents the number of times concept j is answered correctly and immediately after concept i is answered correctly.

Correct transition graph is a directed graph denoted as T. We first calculate the transition probability matrix $\tilde{T}$: $T_{ij} = \frac{C_{ij}}{\sum_k C_{ik}}$ if $i \neq j$; else, it is 0. Here, C is the correct

[6]https://github.com/jennyzhang0215/DKVMN/tree/master/data/assist2015

TABLE I: The statistics of the dataset.

Statistics	ASSISTments	Junyi
# learners	19,840	1,000
# sequence	19,840	59,792
# exercise-performance pair	683,801	4,049,359
# vertexes	100	835
# prerequisite relations	1,112	978
# similarity relations	1,512	1,040

TABLE II: Characteristics of the comparison methods.

	Modeling Concept Relations	Directed	Undirected
BKT	×	×	×
DKT	×	×	×
DKT+	×	×	×
DKVMN	✓	×	×
GKT	✓	✓	×
SKT (ours)	✓	✓	✓

graph. T_{ij} indicates the probability that the influence can be unidirectionally propagated from concept i to concept j. Then, we determine the relations by $T_{ij} = 1$ if $\tilde{T}_{ij} > threshold$; else it is 0, where $threshold$ is set as the average value of $\tilde{T}$ 0.02. Loops are deleted to keep the graph a DAG.

Correct concurrency graph is a undirected graph denoted as O. We first calculate the correct concurrency matrix: $\tilde{C}_{ij} = \frac{C_{ij}+C_{ji}}{|C_{ij}-C_{ji}|+\epsilon}$, where $\epsilon = 0.1$ is used to prevent zero division. And then we use the *max-min-scaling* method to scale $\tilde{C}$ and get $\tilde{O}$: $\tilde{O}_{ij} = \frac{\tilde{C}_{ij}-min(\tilde{C})}{max(\tilde{C})-min(\tilde{C})}$. $\tilde{O}_{ij}$ is the probability that the influence can be bidirectionally propagated between concept i and j. Finally, we determine the relations by $O_{ij} = 1$ if $\tilde{O}_{ij} > threshold$; else it is 0, where $threshold$ is set as the average value of $\tilde{O}$ 0.02.

B. Experimental Setup

1) Data Partition: For each dataset, we divide the learners into training: test = 8:2. We use 90% of the learners' training data to train SKT and use the automl tool nni[7] to apply TPE algorithm [2] to adjust the hyperparameters on the remaining 10% of the data.

2) Framework Setting: We set the size d_e and d_c for embedding matrix as 64, and the size of hidden states d_h as 64. In ASSISTment, the synchronization propagation method is used on the correct concurrency graph and partial propagation method is employed on the correct transition graph. In Junyi, we adopt synchronization propagation method on similarity relations and partial propagation method on prerequisite relations. We respectively set α in Equation (7) as 0.55 in ASSSISTment and 0.45 in Junyi. The discussion for α will be presented in Section V-G. Dropout [30] is used in Equation 10 from the hidden vectors to the output vectors with a drop probability of 0.5.

3) Training Details: We initialize parameters in all networks with *Xavier* initialization [9], which is designed to keep the scale of gradients roughly the same in all layers. The initialization fills the weights with random values in the range of $[-c, c]$ where c=$\sqrt{\frac{3}{n_{in}+n_{out}}}$. n_{in} is the number of neurons feeding into weights, and n_{out} is the number of neurons the result is fed to. We use the Adam algorithm [16] for optimization. The initial learning rate is set to 0.001. Furthermore, we set mini-batches as 16 and max training epoch number as 30. All models are trained on a Linux server with two 2.30GHz Intel(R) Xeon(R) Gold 5218 CPUs and a Tesla V100-SXM2-32GB GPU.[8]

[7]https://github.com/microsoft/nni

[8]The code is available at https://github.com/bigdata-ustc/XKT

C. Baseline Approaches

1) BKT: BKT[9] [7] is a kind of HMMs. Based on the exercise sequences on a specific concept, BKT uses HMM to model the learner's latent knowledge state as a set of binary variables. Although BKT model assumes that mastered knowledge will not be forgotten, factors such as guessing and slipping are still considered.

2) DKT: DKT [25] applies the recurrent neural network model on the exercise performance sequences to estimate the learner's proficiency on each concept (i.e., knowledge state) simultaneously. DKT takes the one-hot performance vector, and outputs a vector representing the learner' proficiency on all concepts, whose elements are all between 0 and 1.

3) DKT+: DKT+[10] [40] is an extended variant of DKT, which aims at solving two major problems in the DKT model. One is that the DKT model fails to reconstruct the observed input and the other one is the predicted performance for DKT model across time-steps is not consistent. By introducing three regularization terms, the authors redefine the loss function of the original DKT model to enhance the consistency in prediction. Specifically, the loss function in DKT+ is $\mathcal{L}' = \mathcal{L} + \lambda_r r + \lambda_{w_1} w_1 + \lambda_{w_2} w_2^2$, where λ_r is for reconstructing the input and λ_{w_1} and λ_{w_2} are for smoothing the transition in prediction. In experiment, we set $\lambda_r = 0.1, \lambda_{w_1} = 0.003, \lambda_{w_2} = 3.0$.

4) DKVMN: DKVMN [41] is another classic model for knowledge tracing. DKVMN has the capability of exploiting the relationships between underlying concepts and directly output the learner's proficiency on each concept. DKVMN has one static matrix called *key*, which stores the knowledge concepts and the other dynamic matrix called *value*, which stores and updates the mastery levels of corresponding concepts. In ASSISTments, for key memory, we set the memory slot size as 20 and memory state dimension as 50. In addition, for value memory, we set the memory slot size as 20 and memory state dimension as 200. In Junyi, we set the memory slot size as 40 and memory state dimension as 200 for key memory. In addition, we set the memory slot size as 40 and memory state dimension as 200 for value memory.

5) GKT: GKT [23] is a GNN-based knowledge tracing method, which only adopts prerequisite relations to construct the knowledge structure. At each time step, GKT will aggregate the states of neighbors to infer the new state, and update the state of not only what is learning currently but also its neighbors. The size of all hidden vectors and the embedding matrix is set as 32.

[9]https://github.com/myudelson/hmm-scalable

[10]https://github.com/ckyeungac/deep-knowledge-tracing-plus

TABLE III: Performance comparison on the KT task.

Dataset	Eval	BKT	DKT	DKT+	DKVMN	GKT	SKT (ours)
ASSISTments	AUC	0.678	0.727	0.728	0.730	0.735	**0.746**
	F1	0.554	0.541	0.572	0.575	0.577	**0.607**
Junyi	AUC	0.831	0.847	0.889	0.890	0.893	**0.908**
	F1	0.760	0.779	0.819	0.817	0.825	**0.835**

TABLE IV: Performance comparison of SKT and its variants.

Model	ASSISTments		Junyi	
	AUC	F1	AUC	F1
SKT_TE	0.710	0.533	0.887	0.824
SKT_Part	0.711	0.548	0.898	0.829
SKT_Sync	0.736	0.579	0.899	0.828
SKT	**0.746**	**0.607**	**0.908**	**0.835**

For better illustration, we summarize the characteristics of these models in Table II.

D. Evaluation Metrics

Same as the previous works [25], [41], we evaluate models from classification perspective. During evaluation, learner's exercise result is defined as a binary value, in which 0 represents incorrect answer as negative sample and 1 represents correct answer as positive sample. Hence, two popular classification metrics, Area Under ROC Curve (AUC) and F1 Score, are adopted to measure the models performance. An AUC score of 0.5 indicates that the model performance is merely as good as random guess and a higher AUC indicates better performance. The F1 score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0. And a higher F1 score indicates a better performance.

E. Experimental Results

1) Performance Comparison: We first compare the overall performance of SKT with baseline models. Results of two datasets on two performance metrics are presented in Table III. we can find that our proposed SKT achieves a better performance than any other baselines both in AUC and F1 in all datasets. Among baselines, we notice that DKVMN and GKT are the best two models, which either model the relations of concepts or explicitly utilize the existing knowledge structures. This observation demonstrates that utilizing the concept relations (i.e., knowledge structure), no matter explicitly or implicitly, does provide additional useful information for estimating learners' knowledge states. Furthermore, with significant promotion, our SKT achieves the best performance by (1) modeling the temporal and spatial effect based on influence propagation; (2) respectively modeling the propagation ways along different relations. This indicates the importance of simultaneously combining temporal information and spatial information and considering the multiple relations among the knowledge structure.

These evidences indicate that considering *transfer of knowledge* during knowledge tracing and modeling the influence propagation with the help of knowledge structure in a proper way can significantly enhance the model effectiveness.

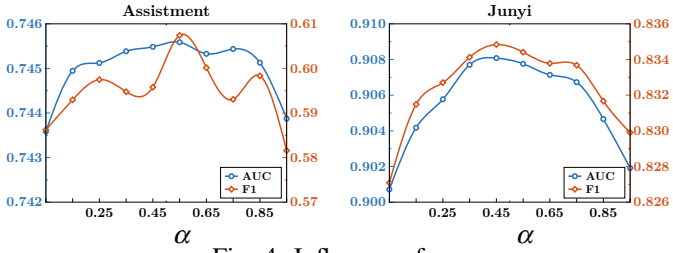


Fig. 4: Influence of α.

F. Ablation Study

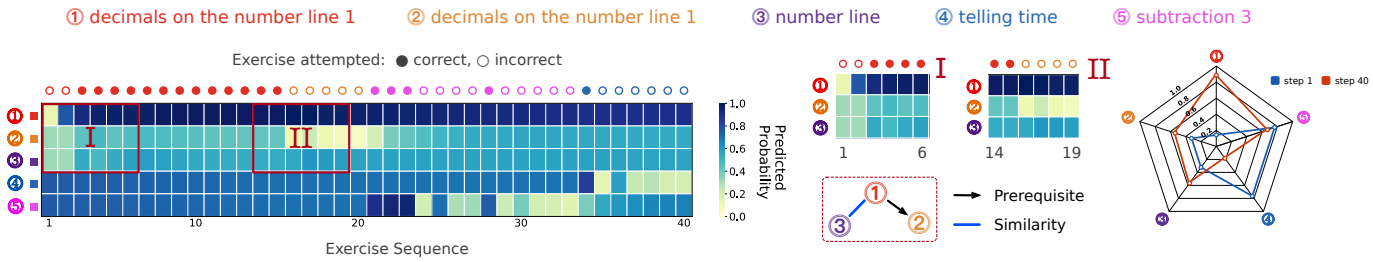
In this part, we compared our models with its variants. SKT_TE, SKT_Part and SKT_Sync are three variants of our model. SKT_TE only models the temporal effect. SKT_Part and SKT_Sync respectively models either partial propagation or synchronization propagation. From Table IV, we can see the two variants (i.e., SKT_Part and SKT_Sync) models the spatial effect have a better performance than SKT_TE which only modeling the temporal effect. This phenomenon suggests that it is important to model the influence propagated in the knowledge structure. Meanwhile, we also observe that SKT have a significant promotion by combining two propagation methods together. This indicates that when we model the influence propagation, it is critical to consider the different ways of the propagation along different relations.

G. Parameter Sensitivity

In SKT, the trade-off parameter α plays a crucial role which balances the contribution from different influences of similarity and prerequisite in Eq. (7). When α is smaller, the influence tends to prioritize the influence from prerequisite relations. Conversely, as α is larger, the model is allowed to focus more on the influence from similarity relations. We perform an experiment on different α where α is selected from $\{0.05, 0.15, ..., 0.95\}$. As shown in Figure 4, when α increases, the performance of SKT increases at the beginning. However, the performance afterwards decreases in all three datasets. These results indicate that properly balancing the influence from prerequisite and similarity relations is vital for achieving more accurate prediction performance.

H. Case Study

Figure 5 shows an example of the evolving knowledge states when a learner learns, where each column represents the proficiency on each concept. From area I, we can obviously see from the divergence of the proficiency at time step 2 and



(a) Time series performance diagram. (b) subsequence digram. (c) radar digram.

Fig. 5: An example of a learner's evolving knowledge states of 5 concepts as she solves 40 exercises of Junyi. In sub-figure (a), concepts are marked in different colors on the left side. The top part indicates the performance at each time step. Sub-figure (b) shows two subsequences which show two different propagation effects. The radar figure (c) on the right shows the proficiency (in the range (0, 1)) of 5 concepts at the beginning (T=0) and the end (T=40).

time step 3 of concept ① (*decimals on the number line 1*), which get promoted at step 3. Meanwhile, the proficiency of concept ② (*decimals on the number line 2*) and concept ③ (*number line*) also gets promoted, where concept ② is a successor with the prerequisite relation while concept ③ is the neighbor with the similarity relation. Furthermore, from area II, at step 16, when the learner gets confused with the concept ②, the proficiency of it decreases. However, the proficiency of the predecessor of the concept ①, ② remains stable. This observation indicates that the influence along prerequisite relations is only unidirectionally propagated in SKT. From these observations, we could see that, owing to the ability of tracing the influence propagation among concepts, SKT is able to provide a better interpretable insight on evolving states for knowledge tracing.

I. Concept Clustering

SKT has the power to cluster related or similar concepts into a same group, which can not only help the educational experts discover the relationship among concepts, but also be helpful for improving curricula arrangement. Following Piech et al. [25], we visualize the concept representation vectors utilizing the T-SNE method [21]. Specifically, we first generate the influence feature vector by $J_{ij} = \frac{y(j|i)}{\sum_k y(j|k)}$, where $y(j|i)$ is the average correctness probability assigned by SKT to exercise j when exercise j is answered correctly at the first time step. Then, we reduce the vector dimension to two-dimension space and then obtain the graph of concepts clustering. As shown in Figure 6, the concepts in the same color is in the same group. The arrow size of the edge indicates connection strength, i.e., cosine distance. For better illustration, we choose 42 concepts and omit those edges with cosine distance smaller than 0.5. From Figure 6, we can see that SKT clusters the concepts into five groups, and the concepts in the same group is quite relevant to a certain knowledge area, which is annotated beside the group. Based on the clustering result, the educational expert can better discover the relationship via the connection strength. Meanwhile, the teachers in the school can also arrange the learners to learn the concepts in the same group for they may be more related and may have positive transfer on each other.

VI. Conclusion

In this paper, we proposed a new knowledge tracing framework, i.e., Structure-based Knowledge Tracing (SKT). By utilizing the knowledge structure, SKT succeeds in modeling transfer of knowledge. Specifically, by concurrently considering the influence propagation in the knowledge structure with learners' exercise performance sequence, SKT is able to estimate learners' knowledge states more precisely. Extensive experiments were conducted on real-world datasets and the results showed the effectiveness and interpretability of SKT.

For the future work, we would try to involve more relations and node attributes in the knowledge structure such as collaboration relations [12]. Besides, we would explore utilizing more features in knowledge tracing along with the knowledge structure such as components in exercises (e.g., equation, image and text). Meanwhile, we would like to apply our SKT on some other educational problems such as cognitive diagnosis assessment [32].

Acknowledgement. This research was supported by grants from the National Natural Science Foundation of China (Grants No. 61922073, 61672483, U1605251) and the Iflytek joint research program. Qi Liu gratefully acknowledges the support of the Youth Innovation Promotion Association of CAS (No. 2014299).

References

[1] J. Atwood and D. Towsley. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*, pages 1993–2001, 2016.

[2] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

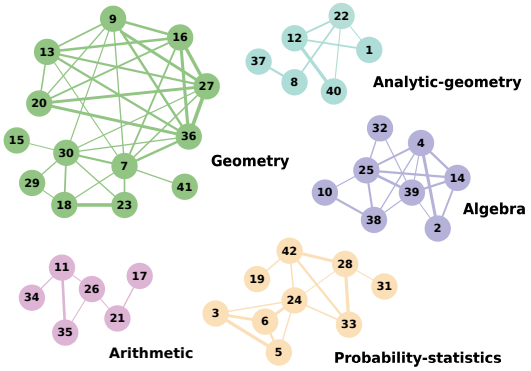
[3] H.-S. Chang, H.-J. Hsu, and K.-T. Chen. Modeling exercise relationships in e-learning: A unified approach. In *EDM*, pages 532–535, 2015.

[4] P. Chen, Y. Lu, V. W. Zheng, and Y. Pian. Prerequisite-driven deep knowledge tracing. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 39–48. IEEE, 2018.

[5] 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. *arXiv preprint arXiv:1406.1078*, 2014.

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.



01 angle types	15 same side exterior angles	29 corresponding angles
02 exponents 3	16 alternate exterior angles 2	30 parallel lines 1
03 reading pictographs 2	17 subtraction 4	31 plotting the line of best fit
04 exponents 2	18 alternate exterior angles	32 multiplying scientific notation
05 reading pictographs 1	19 reading stem and leaf plots	33 reading bar charts 2
06 reading tables 1	20 parallel lines 2	34 addition 2
07 congruent angles	21 subtraction 3	35 subtraction 1
08 vertical angles 2	22 complementary angles	36 same side interior angles 2
09 alternate interior angles 2	23 same side interior angles	37 vertical angles
10 simplifying expressions	24 reading tables 2	38 exponent rules
11 addition 1	25 scientific notation intuition	39 scientific notation
12 supplementary angles	26 subtraction 2	40 complementary angles
13 same side exterior angles 2	27 corresponding angles 2	41 alternate interior angles
14 exponents 1	28 reading line charts 1	42 reading bar charts 1

Fig. 6: Concept clustering.

[8] H. C. Ellis. The transfer of learning. 1965.

[9] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] P. A. Howard-Jones and R. Martin. The effect of questioning on concept learning within a hypertext system. *Journal of Computer Assisted Learning*, 18(1):10–20, 2002.

[12] X. Huang, Q. Liu, C. Wang, H. Han, J. Ma, E. Chen, Y. Su, and S. Wang. Constructing educational concept maps with multiple relationships from multi-source data. In J. Wang, K. Shim, and X. Wu, editors, *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*, pages 1108–1113. IEEE, 2019.

[13] Z. Huang, Q. Liu, Y. Chen, L. Wu, K. Xiao, E. Chen, H. Ma, and G. Hu. Learning or forgetting? a dynamic approach for tracking the knowledge proficiency of students. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–33, 2020.

[14] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[15] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *European conference on principles of data mining and knowledge discovery*, pages 259–271. Springer, 2006.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[17] Y. Li, R. Yu, C. Shahabi, and Y. Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

[18] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1734–1743, 2018.

[19] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang. Exploiting cognitive structure for adaptive learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 627–635, 2019.

[20] Q. Liu, B. Xiang, N. J. Yuan, E. Chen, H. Xiong, Y. Zheng, and Y. Yang. An influence propagation view of pagerank. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 11(3):1–30, 2017.

[21] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[22] K. Nagatani, Q. Zhang, M. Sato, Y.-Y. Chen, F. Chen, and T. Ohkuma. Augmenting knowledge tracing by considering forgetting behavior. In *The World Wide Web Conference*, pages 3101–3107. ACM, 2019.

[23] H. Nakagawa, Y. Iwasawa, and Y. Matsuo. Graph-based knowledge tracing: Modeling student proficiency using graph neural network. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 156–163. ACM, 2019.

[24] J. Piaget. *The equilibration of cognitive structures: The central problem of intellectual development*. University of Chicago press, 1985.

[25] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in Neural Information Processing Systems*, pages 505–513, 2015.

[26] W. F. Pinar, W. M. Reynolds, P. Slattery, and P. M. Taubman. *Understanding curriculum: An introduction to the study of historical and contemporary curriculum discourses*, volume 17. Peter Lang, 1995.

[27] D. H. Schunk. *Learning theories an educational perspective sixth edition*. Pearson, 2012.

[28] Y. Shang, H. Shi, and S.-S. Chen. An intelligent distributed environment for active learning. *Journal on Educational Resources in Computing (JERIC)*, 1(2es):4–es, 2001.

[29] P. R.-J. Simons. Transfer of learning: Paradoxes for learners. *International journal of educational research*, 31(7):577–589, 1999.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[31] E. L. Thorndike and R. S. Woodworth. The influence of improvement in one mental function upon the efficiency of other functions. ii. the estimation of magnitudes. *Psychological Review*, 8(4):384, 1901.

[32] F. Wang, Q. Liu, E. Chen, Z. Huang, Y. Chen, Y. Yin, Z. Huang, and S. Wang. Neural cognitive diagnosis for intelligent education systems. *arXiv preprint arXiv:1908.08733*, 2019.

[33] H. Wang, T. Xu, Q. Liu, D. Lian, E. Chen, D. Du, H. Wu, and W. Su. MCNE: an end-to-end framework for learning multiple conditional network representations of social network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1064–1072, 2019.

[34] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing.

[35] Z. Wang, X. Feng, J. Tang, G. Y. Huang, and Z. Liu. Deep knowledge tracing with side information. In *International Conference on Artificial Intelligence in Education*, pages 303–308. Springer, 2019.

[36] R. S. Woodworth and E. Thorndike. The influence of improvement in one mental function upon the efficiency of other functions.(i). *Psychological review*, 8(3):247, 1901.

[37] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang. A neural influence diffusion model for social recommendation. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 235–244, 2019.

[38] X. Xiong, S. Zhao, E. G. Van Inwegen, and J. E. Beck. Going deeper with deep knowledge tracing. *International Educational Data Mining Society*, 2016.

[39] H. Yang and L. P. Cheung. Implicit heterogeneous features embedding in deep knowledge tracing. *Cognitive Computation*, 10(1):3–14, 2018.

[40] C.-K. Yeung and D.-Y. Yeung. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the Fifth Annual ACM Conference on Learning at Scale*, page 5. ACM, 2018.

[41] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774. International World Wide Web Conferences Steering Committee, 2017.