Automatical Graph-based Knowledge Tracing

Ting Long[1], Yunfei Liu[1], Weinan Zhang[1], Wei Xia[2], Zhicheng He[2], Ruiming Tang[2] and Yong Yu[1]
[1] Shanghai Jiao Tong University, [2] Huawei Noah's Ark Lab
{longting, liuyunfei, wnzhang, yyu}@sjtu.edu.cn, {xiawei24, hezhicheng9, tangruiming}@huawei.com

ABSTRACT

Knowledge tracing (KT) is an essential task in online education, which dynamically assesses students' mastery of concepts by predicting the probability that they correctly answer questions. One of the most effective solutions for knowledge tracing is graph-based methods. They maintain multiple vectors to represent students' mastery of concepts, and use these vectors to predict the probability of students correctly answering questions. To give more accurate predictions, the graph-based methods require concept relations to update these vectors once students answer questions. However, the concept relations usually require manual annotation in a real-world scenario, limiting the application of the graph-based method. In this paper, we proposed a method called Automatical Graph-based Knowledge Tracing (AGKT), which is a graph-based method that updates these vectors without requiring manually annotated concept relations. We evaluate our method on four public datasets and compare it with ten advanced methods. The experiment results demonstrate that AGKT yields superior performance.

Keywords

online learning, knowledge tracing, graph-based method

1. INTRODUCTION

Knowledge tracing is a task that estimates students' mastery of knowledge by predicting the probability that they correctly answer questions. It plays a significantly important role in the online educational application, like exercise recommendation and knowledge diagnosis. The input of knowledge tracing is the question-answering history of a student and a new question, and the output is the probability of the student correctly answering the question. For instance, in step 4, the knowledge tracing aims to use the student's performance on q_1, q_2, q_3 to predict the probability of the student correctly answering q_4 in Figure 1.

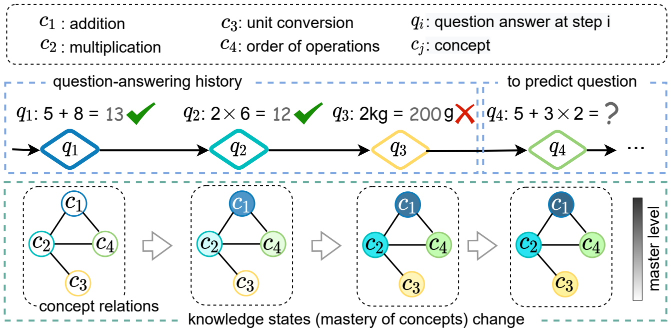


Figure 1: An example of a student's question-answering sequence and the change of her (his) knowledge states. The darkness of a circle's color represents the level of the student masters the corresponding concept, and more darkness denotes a better mastery.

To address the knowledge tracing problem, many outstanding methods [11, 17, 8, 19, 7, 1, 5, 3, 10, 18, 12, 9, 14] have been proposed, which could be grouped into graph-free methods [11, 17, 8, 19, 7, 1, 5, 3, 10, 18, 12] and graph-based methods [9, 14]. Graph-free methods are directly built based on the sequential models, like auto-regressive methods [13, 6, 2], Transformer [15]. They maintain one or multiple vectors to represent students' mastery of knowledge concepts, which denotes as knowledge states, and they predict students' performance based on vectors. As students' knowledge states change with time, to maintain the latest knowledge states for prediction, they update the vectors which represent knowledge states according to the questions and question-concept relation immediately after students answer questions. However, according to the previous research, concepts in one specific domain are correlative with each other [7, 19]. Thus, when a student has a deeper understanding of one concept, her(his) mastery of the correlative concepts also changes. For instance, answering q_1 enhances the student's mastery of *addition* in Figure 1. Since *addition* is correlative with *multiplication* in the concept relation graph, the deeper understanding of *addition* enhances her (his) mastery of *multiplication*. Hence, the knowledge states are not only influenced by questions and question-concept relations, but also influenced by the relations among concepts.

Due to this reason, the graph-based methods introduce the concept-relation graph when they update students' knowl-

T. Long, Y. Liu, W. Zhang, W. Xia, Z. He, R. Tang, and Y. Yu. Automatical graph-based knowledge tracing. In A. Mitrovic and N. Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 710–714, Durham, United Kingdom, July 2022. International Educational Data Mining Society.

https://doi.org/10.5281/zenodo.6853057

edge states. Though the graph-based methods gain many achievements in performance and interpretability, the concept-relation graph usually relies on manual annotation, which is almost impossible for cases with thousands of concepts. To address this issue and keep the superiority of graph-based methods, we proposed a method called Automatical Graph-based Knowledge Tracing (AGKT), which is independent on the manual-annotated concept relation, but could update students' knowledge states like graph-based methods by considering question, question-concept relation and concept-concept relation. We perform experiments on four real-world datasets, and compare our method with ten excellent models. The experiment results reveal that our method has superior performance to the other methods.

2. PROBLEM FORMULATION

Suppose the learning history of a student is $\mathcal{X}_{t-1} = \{(q_1,\ r_1),\ (q_2,\ r_2),\ ...,\ (q_{t-1},\ r_{t-1})\}$. Here, q_i denotes the question the student answers at step i. r_i denotes the correctness of the student's response on q_i, and

$$r_i = \begin{cases} 1, & \text{if the student's answer is right;} \\ 0, & \text{otherwise.} \end{cases} \qquad (1)$$

Each question is related to one or multiple concepts. We denote the set of concepts as $C = \{c_j\}_{j=1}^{|C|}$, and the set of the concepts which are related to q_i as C_i, and the set of the concepts which are unrelated to q_i is denoted as D_i. Obviously, $C_i \cup D_i = C$.

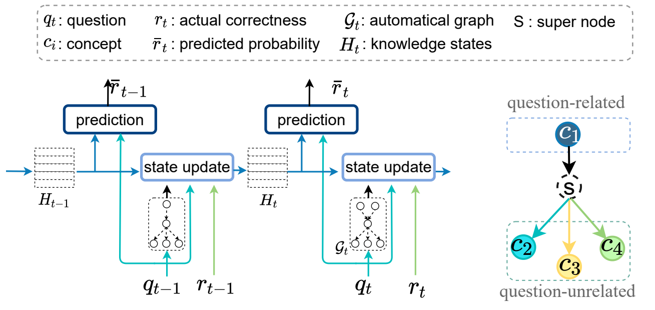
The task of knowledge tracing is formulated as estimating the probability that the student correctly answer a new question q_t given the question-answering history $\mathcal{X}_{t-1}$, *i.e.*, $P(r_t = 1|q_t, \mathcal{X}_{t-1})$. We approach that by learning a function to estimate the probability:

$$\bar{r}_t = f_\Theta(\cdot), \qquad (2)$$

where $\bar{r}_t = P(r_t = 1|q_t, \mathcal{X}_{t-1})$ and the input of $f_\Theta(\cdot)$ represent the features used to predict the correctness of the student.

3. METHOD

As we discussed previously, students' latest knowledge states are necessary to predict the probability of students correctly answering questions. Moreover, considering the question, the question-concept relation and concept-concept relation in knowledge state update benefits for the performance. Nevertheless, annotating the concept-concept relation is almost impossible. To reserve the superiority of the graph-based methods in the condition of manual annotated concept-concept relation absent, we proposed a method called Automatical Graph-based Knowledge Tracing (AGKT). It is composed of three components, as Figure 2 (a) illustrates: Automatical Graph (AG), state update module and prediction module. AG is obtained according to question-concept relation, which replaces the concept relation graph to assist the state update module in updating students' knowledge states. The prediction module predicts the probability of students correctly answering questions by students' knowledge states and the question information. In the following, we will discuss the AG first. Then, we will present how to update students' knowledge states based on AG in the state update module. Finally, we will discuss the prediction module.



(a) The framework

(b) An example of AG

Figure 2: (a) The framework of Automatical Graph-based Knowledge Tracing (AGKT). (b) An example of the Automatical Graph (AG) for the q_1 in Figure 1.

3.1 Automatical Graph (AG)

Each question corresponds to an Automatical Graph (AG). Figure 2(b) illustrates the corresponding AG for q_1 in Figure 1. We denote the graph corresponds to q_t as $\mathcal{G}_t$, and $\mathcal{G}_t = \{\mathcal{V}_t,\ \mathcal{E}_t\}$. $\mathcal{V}_t$ is the node set, and $\mathcal{V}_t = \{V_t^c,\ V_t^d,\ s\}$. Here, each node in V_t^c represents a concept in C_t. That is, each node in V_t^c represents one question-related concept. Each node in V_t^d represents a question-unrelated concept (D_t). For the ease of presentation, we call these nodes by concepts when there is no ambiguity. s is the supernode, which connects the question-related concepts and question-unrelated concepts. $\mathcal{E}_t$ represents the edges on $\mathcal{G}_t$, and $\mathcal{E}_t = \{E_c,\ E_d\}$, E_c represents the edges between V_t^c and s, and E_d represents the edges between V_t^d and s.

3.2 The state update module

Since the questions require students to utilize their knowledge of question-related concepts to answer, which increases their comprehension of the question-related concepts, and causes their knowledge states on these concepts to change once they answer questions. As concepts are correlative, these changes will also influence the knowledge states of some question-unrelated concepts. Based on this, we maintain a vector for each concept to represent a student's knowledge state on it. When a student answer a question, we update the knowledge states of question-related concepts first, and then we update the knowledge states of question-unrelated concepts according to the AG.

3.2.1 The state update of question-related concepts

Suppose a student interacts with question q_t at step t. After the student answer q_t, we update the knowledge states of question-related concepts by concepts' attributes and the student's response on q_t. Specifically, for a concept $c_i \in C_t$, we integrate its embedding and the student's correctness by:

$$\mathbf{z}_t^i = \begin{cases} \mathbf{e}_i^c \ \oplus \ \mathbf{0}, & r_t = 1, \\ \mathbf{0} \ \oplus \ \mathbf{e}_i^c, & r_t = 0, \end{cases} \qquad (3)$$

where $\mathbf{0} = (0, 0, ..., 0)$, is a d dimension zero vector. $\mathbf{e}_i^c \in \mathbf{R}^d$ denotes the embedding of c_i. $\oplus$ denotes the concatenation.

Then, we feed the concatenation into the cell of recurrent neural network (RNN) to update the knowledge state of c_i:

$$\mathbf{h}_{t+1}^i = GRU(\mathbf{z}_t^i, \mathbf{h}_t^i), \tag{4}$$

where $\mathbf{h}_t^i$ denotes the knwoledge states on concept c_i at step t, it is the i-th row of the knowledge states matrix $\mathbf{H}_t$. GRU denotes the gated recurrent unit [2].

3.2.2 *The state update of question-unrelated concepts.*

The knowledge states of one concept will influence the concepts that correlate to it, *e.g.*, the knowledge state of *multiplication* is influenced by the knowledge state of *addition* after the student answer q_1 in Figure 1. Thus, the update of question-unrelated concepts should be based on the relation of concepts. Since the concept relation is unknown in many cases, we update the knowledge states of question-unrelated concepts based on AG (Figure 2 (b)). Since the question-related concepts connect to the supernode, and the supernode connects the question-unrelated concepts in AG, the supernode integrates the message about the knowledge states of all the question-related concepts first. Then it transmits the integrated message to question-unrelated concepts. Finally, the question-unrelated concepts update their knowledge states according to the message transmitted from supernode. Specifically, we take the following steps to update the knowledge states of the question-unrelated concepts:

First, we represent the original message that a question-related concept c_i sent to the supernode by:

$$\mathbf{m}_t^i \;=\; \mathbf{h}_{t+1}^i \;\oplus\; \mathbf{o}_i, \tag{5}$$

where $\mathbf{o}_i$ is the one-hot encoding of concept c_i to denote the identity of the message source. The supernode aggregates the messages from the question-related concepts by:

$$\mathbf{h}^s \;=\; \frac{1}{|C_t|} \sum_{c_i \in C_t} f_s(\mathbf{m}_t^i), \tag{6}$$

where f_s denotes multi-layer perceptrons (MLP). $\mathbf{h}^s$ denotes the states of the supernode.

Then, the supernode transmits the message it receives to the question-unrelated concepts. We represent the message obtained by the question-unrelated concept c_j $(c_j \in D_t)$ as

$$\mathbf{m}_t^j \;=\; f_{d,j}(\mathbf{h}_s), \tag{7}$$

$f_{d,j}$ denotes the mapping function, which is implemented by MLP, to extract the message which is interested by c_j, *e.g.*, the knowledge states of its correlative concepts. Note, for different question-unrelated concepts, the mapping functions in Eq. 7 are different. However, for the same question-unrelated concept in different time steps, the mapping function is shared to guarantee the pattern are compatible in all steps.

Finally, we update the knowledge state on question-unrelated concept c_j by:

$$\mathbf{h}_{t+1}^j = GRU(\mathbf{m}_t^j, \mathbf{h}_t^j). \tag{8}$$

Table 1: Dataset statistics.

Dataset	ASSIST09	ASSIST12	EdNet	Junyi Math
Students	2,968	22,422	50,000	1,146
Records	185,110	1,839,429	3,266,010	101,854
Questions	15,003	45,543	12,077	1,145
Concepts	121	99	189	39
Questions Per Concept	150.76	460.03	144.01	36.28
Concepts Per Question	1.22	1.0	2.67	1.0
Attempts Per Question	12.34	40.39	270.43	71.98
Attempts Per Concept	1,914.21	18,580.10	39,959.14	2,611.64
Positive Label Rates	63.80%	69.60%	59.54%	66.78%

3.3 The prediction module

To predict the probability of students correctly answering questions, we consider the attributes of question q_t, the level of the student master question-related concepts, and her(his) knowledge background. We represent the attributes of question q_t by the question embedding $\mathbf{e}_t^q$, and $\mathbf{e}_t^q \in \mathbf{R}^d$. We represent the level of the student master the question-related concepts by the mean knowledge state:

$$\mathbf{h}_m \;=\; \frac{1}{|C_t|} \sum_{c_i \in C_t} \mathbf{h}_t^i. \tag{9}$$

We represent students' knowledge background as

$$\mathbf{h}_c \;=\; \frac{1}{|C|} \sum_{c_i \in C} \mathbf{h}_t^i, \tag{10}$$

and then we predict the probability of the student correctly answering q_t by

$$\bar{r}_t \;=\; \delta(f_r(\mathbf{h}_c \oplus \mathbf{e}_t^q \;\oplus\; \mathbf{h}_m), \tag{11}$$

where f_r denotes MLP, and δ denotes the Sigmoid function.

3.4 Model Learning

The objective function of our model is to minimize the negative log-likelihood of the observed sequence. The sequence is the question-answering history of the student from step 1 to T. The learning parameters of our method are the embedding of concepts and questions, the weights in GRU, the parameters of all the MLPs. The parameters are jointly learned by minimizing the cross-entropy between the predicted probability $\bar{r}_t$ and the students' actual correctness r_t as

$$\mathcal{L} \;=\; -\sum_{i=1}^{T}(r_i \; log \; \bar{r}_i \;+\; (1-r_i) \; log(1-\bar{r}_i)). \tag{12}$$

4. EXPERIMENT

4.1 Dataset

We evaluate our method on four public datasets: ASSIST09, ASSIST12, EdNet, and Junyi Math. These datasets record the question-answering history of students. We take the questions, the concepts related to the questions, and the students correctness of responses from the records. The maximum length of students' question-answering history is set to 200. We split 80% data for training and validation, and 20% for testing. The statistics of the four datasets are shown in Table 1. Note, the statistics are the actual samples we use in our experiments after preprocessing, which are different from the statistics of raw data.

4.2 Baselines

To evaluate the effectiveness of our model, we compare our method with graph-free methods, and graph-based methods. The graph-free methods trace students' knowledge

Table 2: The AUC on four public datasets.

Model	ASSIST09	ASSIST12	EdNet	Junyi Math
BKT	0.6815	0.6142	0.5729	0.6293
KTM	0.6734	0.6881	0.7071	0.7207
SAKT	0.6884	0.6914	0.7313	0.7422
SAINT	0.6901	0.6917	0.7336	0.8079
AKT-NR	0.6940	0.7098	0.7450	0.7705
DKT	0.6769	0.6884	0.7496	0.8397
DHKT	0.7499	0.6966	0.7547	0.8594
EERNNA	0.7244	0.7000	0.7437	0.8196
DKVMN	0.7235	0.6664	0.7009	0.8426
GKT	0.7488	0.6857	0.7039	0.8257
AGKT	**0.7765**	**0.7232**	**0.7589**	**0.8656**

states without considering concept-relation, and they could be grouped into four groups. The first group are traditional methods, *i.e.*, BKT [4], KTM [16], which trace the students' knowledge states according to the factors that affect students' learning. The second group are single-state methods, *i.e.*, DKT [11], DHKT [17], EERNNA [7], which use one vector to represent students' knowledge states on all concepts. The third group is multi-state methods, *i.e.*, DKVMN [19], which maintain a vector for each concept to represent students' knowledge states on them. The fourth group are state-free methods, *i.e.*, SAKT [10], SAINT [3], AKT-NR [5], which maintain no explicit vector to represent students' knowledge states. The graph-based methods apply a concept-relation graph in the knowledge state update. Here we choose the GKT [9].

4.3 Student Response Prediction

We measure the AUC to evaluate the performance of models. A higher AUC indicates a better performance. Table 2 shows the converged ACC, AUC. According to Table 2, on the ASSIST09, our method is better than the best baseline DHKT by 2.66% in AUC. On ASSIST12, our model outperforms the best baseline AKT-NR 1.34% in AUC. On EdNet, our model is better than the best baseline DHKT by 0.42% in AUC. On Junyi, our method is better than the best baseline DHKT 0.64% in AUC. Thus, the performance presented in Table 2 demonstrates that our method is effective.

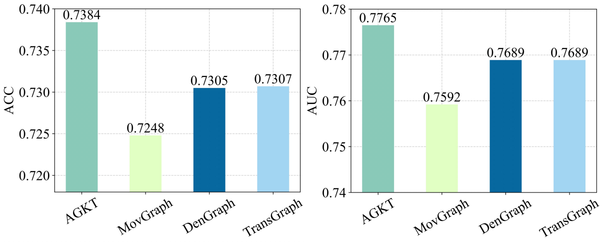
4.4 Ablation Study

To further verify the contribution of AG, we conduct experiments on ASSIST09 with three comparative settings:

- **MovGraph** removes the AG from AGKT. Thus, this setting removes the concept-concept relation.
- **DenGraph** replaces the AG in AGKT with the dense graph in GKT [9]
- **TransGraph** replaces AG with the transition graph in GKT.

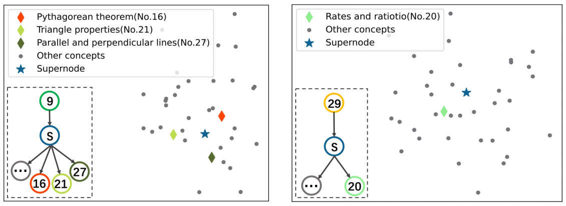
The experimental results are shown in Figure 3. We can find that: (1) *AGKT*, *DenGraph* and *TransGraph* have better performance than *MovGraph*. That means the consideration of concept-concept relation is beneficial; (2) Replacing AG in AGKT with other graphs (DenGraph, TransGraph) decreases the performance. That means the AG is more effective in the framework of AGKT.

4.5 Case Study



(a) ACC. (b) AUC.

Figure 3: The contribution of AG.



(a) Congruent triangles (No.9). (b) Ratio percentage (No.29).

Figure 4: The state of supernode and the message received by the question-unrelated concepts.

To investigate whether the states update of question-unrelated concepts is interpretable, we randomly pick two question-answering records, and visualize the state of supernode and the message obtained by question-unrelated concepts. We expect the knowledge states of question-unrelated concepts could be automatically updated according to the actual relationship with the question-related concepts when AGKT converges. Thus, for the correlative concepts of the question-related concepts, the message they receive from supernode should be correlated with the state of the supernode.

The results is presented in Figure 4. We can observe the message obtained by *triangle properties* (No.21) is highly close to supernode when the student answer the question related to *congruent triangles* (No.9), and message obtained by *rates and ratios* is highly close to the state of supernode when the student answer the question related to *ratio-percentage* (No.29). Thus, the state update of question-unrelated concepts is interpretable in our method. which demonstrates our method has good interpretability.

5. CONCLUSION

In this paper, we proposed a method called Automatical Graph-based Knowledge Tracing (AGKT). Different from the previous graph-based methods, it adopts the Automatical Graph (AG) to automatically update students' knowledge state without requiring the manually annotated concept relation. We evaluate the performance of our method on four public real-world datasets, and compare it with ten methods. The experiment result reveals that our method is effective in tracing student's knowledge states.

6. ACKNOWLEDGMENTS

The corresponding author Yong Yu is supported by National Natural Science Foundation of China (62177033). The work is also sponsored by Huawei Innovation Research Program.

References

[1] G. Abdelrahman and Q. Wang. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 175–184, 2019.

[2] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[3] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning@ Scale*, pages 341–344, 2020.

[4] 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.

[5] A. Ghosh, N. Heffernan, and A. S. Lan. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2330–2339, 2020.

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

[7] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):100–115, 2019.

[8] 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, 2019.

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

[10] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[11] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. volume 28, pages 505–513, 2015.

[12] 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 *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1857–1860, 2020.

[13] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. *arXiv preprint arXiv:1503.08895*, 2015.

[14] S. Tong, Q. Liu, W. Huang, Z. Huang, E. Chen, C. Liu, H. Ma, and S. Wang. Structure-based knowledge tracing: An influence propagation view. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 541–550. IEEE, 2020.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[16] J.-J. Vie and H. Kashima. Knowledge tracing machines: Factorization machines for knowledge tracing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 750–757, 2019.

[17] T. Wang, F. Ma, and J. Gao. Deep hierarchical knowledge tracing. In *12th International Conference on Educational Data Mining, EDM 2019*, pages 671–674. International Educational Data Mining Society, 2019.

[18] S. Yang, M. Zhu, J. Hou, and X. Lu. Deep knowledge tracing with convolutions. *arXiv preprint arXiv:2008.01169*, 2020.

[19] 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, 2017.