Grasping or Forgetting? MAKT: A Dynamic Model via Multi-head Self-Attention for Knowledge Tracing

Deming Sheng, Jingling Yuan*, Xin Zhang
School of Computer Science and Technology,
Wuhan University of Technology,
Wuhan 430070, China
Email: shengdeming@whut.edu.cn, yjl@whut.edu.cn, xinz@whut.edu.cn

***Abstract*—The outbreak of the COVID-19 pandemic arises enormous attention to online education then knowledge tracking is an increasingly crucial task with its vigorous development. However, the surge of student historical interactions and the lack of prior knowledge is engendering a sequence of issues, such as the decrease in prediction accuracy while the increase in training time. Simultaneously, most existing approaches fail to provide in-depth insights into why a student is likely to answer the question incorrectly and what affects the knowledge state of the student. To address those issues, we propose a multi-head self-attention model named MAKT for dynamic knowledge tracing, which makes the prediction results interpretable at the model and instance level. The customized multi-head self-attention layer has high training efficiency owing to its parallelization capability and spends about 6 seconds in each epoch on a single GPU. We further visualize the attention weights of MAKT and student knowledge acquisition tracking, finding that not all historical interactions are equally important but the recent interactions profoundly establish the knowledge state of students. In the end, extensive experiments on three datasets demonstrate the robustness and superiorities of MAKT, improving ACC by 1.14 % and AUC by 1.20 % on average.**

***Index Terms*—MOOC, Knowledge Tracing, Educational Data Mining, Attention Mechanism, Sequence Modeling**

I. Introduction

Online education systems, such as Massive Online Open Course (MOOC), Intelligent Tutoring System (ITS) and Online Judge (OJ) Systems, have a long history dating back to the 1980s [1], [2] and have still witnessed the proliferation with the computer-aid technology and artificial intelligence in recent years. Specifically, students in these systems can finish a series of appropriate tests individually according to their needs and acquire the necessary knowledge in the process of solving relevant exercises. As shown in Fig.1, the availability of such exercising process offers an opportunity to model student learning in terms of predicting student performance (e.g., forecasting whether or not this student can answer an exercise correctly in the next time) and tracking student knowledge state (e.g., estimating the mastery level of key knowledge components based on historical data.).

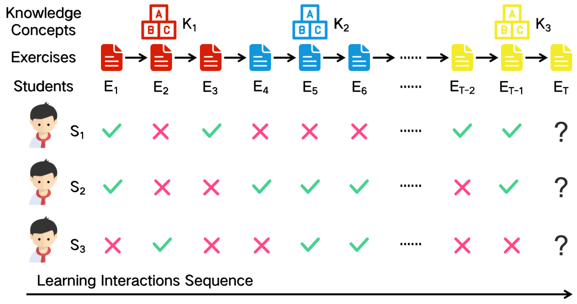

Knowledge Concepts
A B C K₁
A B C K₂
A B C K₃
Exercises
Students
E1 E2 E3 E4 E5 E6 ...... ET-2 ET-1 ET
S1 ✓ ✗ ✓ ✗ ✗ ✗ ...... ✓ ✓ ?
S2 ✓ ✗ ✗ ✓ ✓ ✓ ...... ✗ ✓ ?
S3 ✗ ✓ ✗ ✗ ✓ ✓ ...... ✗ ✗ ?
Learning Interactions Sequence

Fig. 1. An example of the learning process.

Knowledge tracing has undergone many paradigm shifts in the past forty years and many approaches have been developed from both educational psychology and data mining areas, such as sparse factor analysis [3], deep learning [4], topic modeling [5] and matrix factorization [6]. Hidden Markov Model (HMM) was traditionally utilized in Bayesian Knowledge Tracing (BKT) and its variants [7]. More recently, a series of Recurrent Neural Network (RNN) based sequential models have been proposed to capture these long term dependencies between the student historical interactions, such as DKT [4] and DKT+ [8]. Simultaneously, Convolutional Neural Network (CNN) is gradually employed to model individualization in the student learning process [9].

Nonetheless, there are still three main challenges in the knowledge tracking task: (1) Long sequence information modelling and (2) Hidden relationship mining between exercises and (3) Interpretation of the prediction results. Existing approaches have achieved certain results in the first two points, but failed to provide in-depth insights into

DOI reference number: 10.18293/SEKE2021-031

why a student is likely to answer the question incorrectly and **what** affects the knowledge state of the student.

In this paper, we propose a Multi-head Self-Attention model for Knowledge Tracing (MAKT). MAKT can effectively improve the predicted performance and dynamically track the knowledge state. More importantly, MAKT has excellent interpretability and the potential to exploit the implicit relationship between exercises without prior knowledge. In summary, our main contributions in this paper are three folds:

- We customize a multi-head self-attention layer to model individualization, positional encoding rather than the traditional RNN-based model is utilized to capture sequence information.
- We perform extensive experiments on three different datasets and demonstrate that MAKT in addition to showing its robustness and superiorities, supports parallel computing.
- We visualize attention weights and student knowledge acquisition tracking, offer intuitive and in-depth insights on the predicted result at both the model and instance level.

II. Related Work

Cognitive diagnosis refers to predict student performance by discovering student states from the exercising records in educational psychology. The traditional cognitive diagnostic models can be divided into two groups: continuous models and discrete models. Taking item response theory (IRT) as an example of the continuous model, IRT utilizes the logistic regression based on the student ability and the exercise (item) difficulty to assume student performance [10]. Discrete models, such as Deterministic Inputs, Noisy-And gate model (DINA), leverage the student knowledge components proficiency by a binary latent vector with a given Q-matrix to improve prediction results [11].

Knowledge Tracing is an essential task for evaluating the knowledge state of a student based on his past interaction. Bayesian knowledge tracing (BKT), followed by Hidden Markov Model (HMM), models the latent knowledge state as a set of binary variables to trace it. Further extensions incorporate more side information about student's prior knowledge and exercise difficulty into BKT [7]. More recent approaches leverage factorization methods to model individualization with a latent vector that depicts student's knowledge state [6].

Another line of research includes methods based on Deep Learning, which has achieved great success. Deep Knowledge Tracing (DKT) [4] employs Long Short Term Memory (LSTM) to model student exercising process while DKT+ [8] exploits a regularization term on the foundation of DKT to further improve the predicted performance. Memory Augmented Recurrent Neural Network (DKVMN) [12] is proposed to bridge the gap between exercises and knowledge concepts for a better performance prediction. CKT [9] utilizes a hierarchical convolutional network to model individualization.

TABLE I
NOTATIONS

Notations	Description
b	The bias vector
i	The i-th dimension of embedding
$r, \hat{r}$	The actual and predicted label
$\mathbb{E}$	The latent embedding matrix
$\mathbb{Q}$	The knowledge matrix
D	The dimension of latent embedding
E	The total number of exercises
H	The number of Heads
N	The number of Encoders
S	The total number of students
T	The time of learning sequence
W	The weight matrix
X	The input of layer
Q, K, V	The query, key and value matrix

III. The Proposed Model

A. Problem Definition

In an online education system, suppose there are S students, E exercises, and K knowledge concepts, where students do these exercises individually at different times. As shown in Fig.1, the knowledge tracing (KT) task can be formalized as follows: given a learning sequence $x_s = \{(e_1, k_1, r_1), (e_2, k_1, r_2), ..., (e_T, k_K, r_T)\}$ or $x_s = \{(e_1, r_1), (e_2, r_2), ..., (e_T, r_T)\}$ with T learning interactions of a certain student s, we aim to assess the knowledge state of students after each learning interaction. Here e_t represents the exercise being answered at learning interaction $t \in T$ and $r_t \in \{0, 1\}$ indicates whether the exercise e_t has been answered correctly (1 stands for right and 0 else). In short, knowledge tracing aims to estimate the probability $P[r_t = 1|(e_1, k_1, r_1), ..., (e_{T-1}, k_K, r_{T-1}), e_T]$ or $P[r_t = 1|(e_1, r_1), ..., (e_{T-1}, r_{T-1}), e_T]$.

In the following, we will specify the probabilistic modelling and parameter learning of MAKT. For better illustration, the key notations are summarized in Table I.

B. Model of MAKT

The framework we propose approach is showed in Fig.2. The core part of our framework is the multi-head self-attention layer, which utilizes the attention mechanism to better model the learning process of students.

Input Embedding. We firstly transform learning sequence of student into an interaction embedding matrix $\mathbb{E}^{S \times 2D}$, where $2D$ is the latent dimension. Following [13], we extend the answer value r_t to a zero vector e_r = (0, 0, ..., 0) with the same D dimensions as the

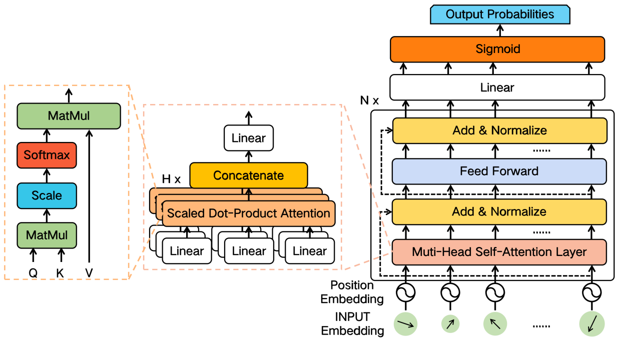


Fig. 2. An illustration of the proposed model.

exercise embedding e_k and integrate them into the input embedding as follows:

$$x = \begin{cases} [e_r \bigoplus e_k], & if \quad r_t = 0 \\ [e_k \bigoplus e_r], & if \quad r_t = 1 \end{cases} \tag{1}$$

Considering the actual situation of different datasets, the original information is distinct, and we adopt two methods to initialize e_k in this paper. If one exercise relates to one knowledge concept or more, we can construct a binary knowledge matrix $\mathbb{Q}^{E \times K}$. However, The number of feature categories is massive, resulting in the generated matrix being high-dimensional and sparse. Hence, we employ an embedding layer to reduce dimensionality and reshape it into a D-dimensional space ($\mathbb{Q}^{E \times K} \rightsquigarrow \mathbb{E}^{E \times D}$). The other method is to randomly initialize $\mathbb{E}^{E \times D}$ with an embedding layer, for it will be updated automatically in the later training process.

Position Embedding. We do not utilize the recurrent and convolution units but the positional encoding to capture sequence information, for the positional encoding is superior in long-distance feature capture capability and operational efficiency. There exist multiple options for position coding [14] and the widely adopted one can be formulated as follows:

$$\begin{aligned} PE(t, 2i) &= sin(t/10000^{i/D}) \\ PE(t, 2i+1) &= cos(t/10000^{i/D}) \end{aligned} \tag{2}$$

Where t is the absolute sequence of each interaction and i is the i-th dimension of the input embedding x. The adopted positional encoding sine and cosine functions have periodicity. For a fixed-length deviation Δ, $PE_{t+\Delta}$ can be expressed as a linear change of PE_t, which is convenient for the model to learn a relative sequence relationship between interactions.

Self-Attention. We employ the scaled dot-product attention mechanism [15] rather than additive attention, for this attention mechanism is more computationally efficient and space-saving. The calculation process of self-attention is as follows:

$$\begin{aligned} &Q = W_Q(x+e_p), K = W_K(x+e_p), V = W_V(x+e_p) \\ &Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \end{aligned} \tag{3}$$

where Q, K, V represent the query, key and value matrix, $W_* \in \mathbb{R}^{D \times d_*}$ is the corresponding weight matrix and $d_* = D/H$. The purpose of scaling through$\sqrt{d_k}$ is to avoid too large dot product because when the dot product is too large, the gradient through $softmax$ will be small. And $softmax$ facilitates the gradient calculation of back propagation, and smooth the result to the [0, 1] interval.

Multi-head Attention. In order to better satisfy parallel calculation while learning different aspects of attention in different subspaces, the attention weight of H (Following [15], we set H to 8) head will be calculated as follows:

$$\begin{aligned} &Multi-head(Q, K, V) = Concat(head_1, ..., head_H)W^O \\ &head_j = Attention(Q_j, K_j, V_j), \ 1 \leq j \leq H \end{aligned} \tag{4}$$

Where $W^O \in \mathbb{R}^{HD \times d_k}$ is the corresponding weight matrix.

Add & Norm. The Add & Norm layer is composed of Add and Norm. Inspired by ResNet [16], Add is a residual connection, usually used to solve the problem of multi-layer network training, allowing the network to focus only on the current difference. Norm refers to layer normalization [17], usually utilized in the RNN structure. Layer normalization converts the input of each layer of neurons into the same mean and variance, which can speed up the convergence. The calculation formula is as follows:

$$\begin{aligned} X &= LayerNorm(X + Multi-head(X)) \\ X &= LayerNorm(X + FFN(X)) \end{aligned} \tag{5}$$

Feed Forward Network. The encoder block consists of sequentially aligned N copies (Following [15], we set N to 6) of encoder layers. A single encoder layer is a multi-headed self-attention layer followed by a feed forward network (FFN) which is defined by:

$$FFN(X) = ReLU(W_1X + b_1)W_2 + b_2 \tag{6}$$

where W_1, W_2 and b_1, b_2 are weights and biases, respectively. In the end, a fully connected network with sigmoid activation is leveraged to obtain the final probability $\hat{r}_t$ of the student:

$$\hat{r}_t = Sigmoid(WX + b) \tag{7}$$

TABLE II
STATISTICS OF ALL DATASETS

DataSet	Statistics				Density
	Students	Concepts	Records	Avg.length	
ASSIST2009	4,151	110	325,637	78	0.71
STATICS2011	333	1,223	189,297	568	0.46
Synthetic-5	4,000	50	200,000	50	1.00

C. Objective Function

To learn all parameters in MAKT and the input embedding matrix $\mathbb{R}^{S \times 2D}$ in the training process, the objective function is to minimize the negative log likelihood of the observed sequence of student responses. We employ the cross-entropy loss between the prediction $\hat{r}_t$ and actual label r_t with the Adam optimizer [18]:

$$\mathcal{L} = -\sum_{t=1}^{T}(r_t log(\hat{r}_t) + (1 - r_t)log(1 - \hat{r}_t)) \tag{8}$$

IV. EXPERIMENTS

A. Data Insights

We adopt two real-world public datasets and one synthetic dataset to show the effectiveness of MAKT. Table II shows the statistics of all datasets.

ASSIST2009: ASSIST2009 is obtained from an online tutoring system named ASSISTments, which covers student response records in 2009 [19]. ASSISTments provides high school math study trajectories, but the original version contains some repeated records, which will make the experimental results less reliable [20]. Hence, the updated version is utilized in this paper and the number of skills reduces from 113 to 110. The updated dataset contains two different versions of skill-builder and non-skill-builder and we adopt the former, which makes the preprocessing of knowledge tracing task more convenient.

STATICS2011: STATICS2011 is collected from engineering mechanics courses in a college [12]. Each exercise contains multiple problem-solving steps in the strength study concept. Since there are fewer different exercises in this dataset, we regard the problem names and the problem-solving step names as skills.

Synthetic-5: Synthetic-5 is a simulation dataset to imitate virtual students learning virtual concepts in 2015 while many works have proved that this dataset is well structured [4]. It is worth noting that each problem has no specific actual concept and each exercise is simulated from five hidden knowledge concepts, also containing the structural relationship between the concepts, the degree of exercise difficulty and the factors that contribute to the growth of knowledge structure in the student learning process.

B. Comparison methods

To illustrate the effectiveness of MAKT, we compare our model with many other models as follows:

DKT [4]: DKT is a deep learning method that utilizes a simple recurrent neural network (RNN or LSTM) to model the exercising process for prediction. We select an LSTM architecture and consider each unique exercise id as a concept associated with the exercise.

DKT+ [8]: DKT+ leverages a regularization term based DKT to enhance the consistency in prediction, which effectively alleviates the two problems in DKT. One is that DKT fails to reconstruct the observed input. The other is the predicted performance for knowledge components across time-steps is not consistent.

DKVMN [12]: DKVMN is a Memory Augmented Recurrent Neural Network where in the relation between different knowledge components are assumed by a key matrix and the student proficiency of each knowledge component by a value matrix.

CKT [9]: CKT is a Convolutional Knowledge Tracing method to model individualization. CKT measures the prior knowledge from the historical learning interactions and utilizes a hierarchical convolutional layer to extract individualized learning rates based on continuous learning interactions of students.

C. Evaluation Metrics

For providing robust evaluation results, the performance was evaluated in terms of Accuracy (ACC) and Area Under Curve (AUC), which widely adopted in the binary classification task. Generally, a larger ACC and AUC value demonstrate better performance.

D. Experimental Results

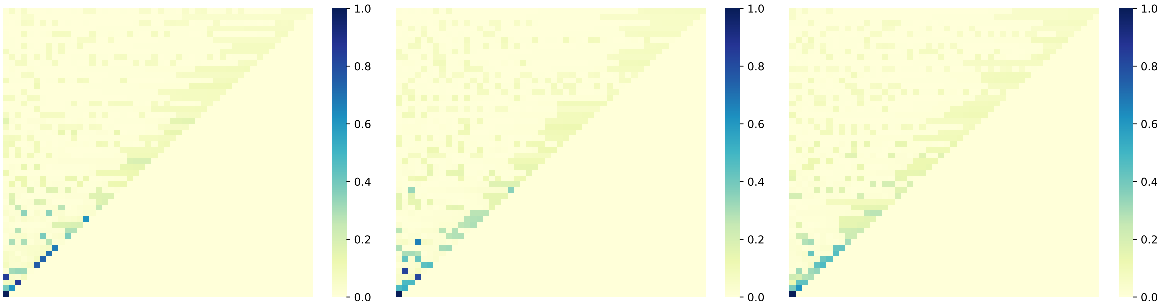
Student Performance Prediction: The performance comparison results on three datasets are shown in Table III. We use **bold** to mark the best performance and underline to indicate the best performance other than MAKT. We can observe that MAKT consistently outperform other baseline models on all datasets, which demonstrates the robustness and superiorities of MAKT. Additionally, MAKT gains higher promotions on dataset ASSIST2009 and STATICS2011 with the longer learning sequence length, which indicates that MAKT can capture the core interactions without falling into certain local irrelevant interactions. For the Synthetic-5 dataset, we suspect that a possible reason for the low improvement is that since the number of knowledge concepts in Synthetic-5 is fairly small (five virtual concepts), this hidden relationship between exercises is not distinguishable and MAKT only leverages the sequence relationship modelled by its self-attention mechanism.

Visualization of Attention Weights: Benefiting from the attention mechanism, MAKT can offer an intuitive

TABLE III
RESULTS OBTAINED WITH DIFFERENT MODELS USING THERE DATASETS.

Datasets	ACC					AUC					%Improv.
	DKT	DKT+	DKVMN	CKT	MAKT	DKT	DKT+	DKVMN	CKT	MAKT	
ASSIST2009	0.7721	0.7734	0.7632	<u>0.7761</u>	**0.7878**	0.8215	0.8234	0.8112	<u>0.8256</u>	**0.8384**	1.5075 // 1.5504
STATICS2011	0.8127	0.8129	0.8113	<u>0.8156</u>	**0.8286**	0.8273	0.8287	0.8275	<u>0.8304</u>	**0.8453**	1.5939 // 1.7943
Synthetic-5	0.7511	0.7523	0.7525	<u>0.7542</u>	**0.7563**	0.8254	0.8262	<u>0.8284</u>	0.8278	**0.8297**	0.2784 // 0.1569
Average	0.7786	0.7795	0.7757	<u>0.7820</u>	**0.7909**	0.8247	0.8261	0.8284	<u>0.8279</u>	**0.8378**	1.1381 // 1.1958

Fig. 3. Visualization of attention weights on different datasets.



(a) ASSIST2009 (b) STATICS2011 (c) Synthetic-5

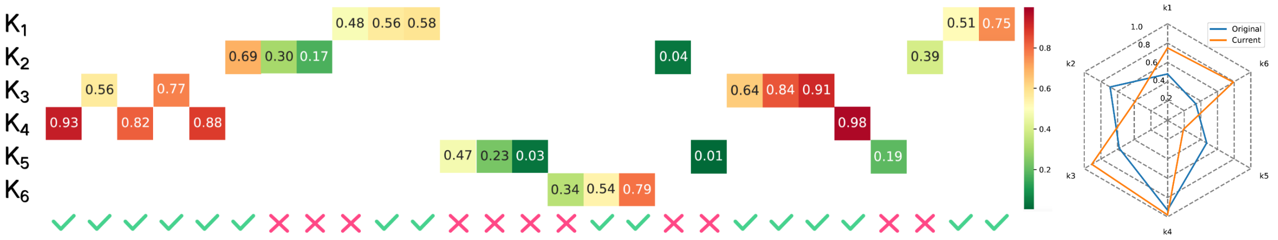


Fig. 4. An example of individualized knowledge tracing result of student.

and in-depth insight on the prediction result with the attention weights visualization, which makes the learning process interpretable at the model level. Fig.3 shows the heatmap of the attention weight matrix on three datasets, each small block depicts the average attention weights of different interaction. An interesting observation is that not all historical interactions are extremely important and the higher weight parameters blocks are concentrated towards the diagonal of the matrix, which can be explained by the forget behaviour rule of the student learning process, that is, the recent interactions profoundly establish the knowledge state of students. Simultaneously, a considerable number of blocks with high attention parameter weights are still scattered in the matrix. Combining with three different datasets, we further find that these interactions share the same knowledge concept with the final interactions, which can be identified by MAKT. A more inspiring conclusion is that the attention mechanism can dig out the hidden relationship between a series of exercises through their attention weights, which benefits the construction of Knowledge Graph in the real world.

Visualization of Knowledge Acquisition Tracking: To make an intuitive and in-depth insight at the instance level, we visualize the predicted mastery levels (i.e., calculated by Eq.(7)) of an exemplified student with the attached knowledge concepts at each interaction during the exercising process. For better visualization, we filter the six most frequent knowledge concepts rather than distinguishing each specific exercise. As shown in Fig.4, we can notice that the current knowledge state is related to both the original knowledge state and the recent interactions. MAKT can dynamically obtain the knowledge state of the student based on his historical data, which is considered meaningful for further online education auxiliary applications in the real world.

Training efficiency: Comparing the other baseline methods, the computational efficiency of MAKT is extremely competitive under the same condition. As shown in the Table IV, MAKT only spends about 6 seconds in each epoch on a single GPU which is 11.7 less than the time taken by DKT+, 7 times less than the time

TABLE IV
Training efficiency comparison of different models on the ASSIST2009 dataset.

	DKT	DKT+	DKVMN	CKT	MAKT
CPU	605	969	344	181	82
GPU	42	70	29	14	6

taken by DKT, 4.8 times less than the time taken by DKVMN and 2.3 times less than the time taken by CKT. Similarly, MAKT outperforms these models on a single CPU because of its parallelization capability.

V. Conclusion

In this paper, we propose a multi-head self-attention based model named MAKT for dynamic knowledge tracing. Specifically, MAKT leverages the historical learning interactions to effectively predict student performance on future exercises and dynamically track the student knowledge state. Simultaneously, MAKT has excellent interpretability and high training efficiency owing to the multi-head self-attention layer, which can offer insights from different levels and support parallel computing. In the end, extensive experimental results demonstrate that MAKT outperforms other baseline models in both ACC and AUC metrics on three different datasets, which indicates the robustness and superiorities of MAKT.

References

[1] M. Yazdani, "Intelligent tutoring systems survey," *Artif. Intell. Rev.*, vol. 1, no. 1, pp. 43–52, 1986.

[2] J. Rickel, "Intelligent computer-aided instruction: a survey organized around system components," *IEEE Trans. Syst. Man Cybern.*, vol. 19, no. 1, pp. 40–57, 1989.

[3] A. S. Lan, C. Studer, and R. G. Baraniuk, "Time-varying learning and content analytics via sparse factor analysis," in *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, S. A. Macskassy, C. Perlich, J. Leskovec, W. Wang, and R. Ghani, Eds. ACM, 2014, pp. 452–461.

[4] 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 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 505–513.

[5] W. X. Zhao, W. Zhang, Y. He, X. Xie, and J. Wen, "Automatically learning topics and difficulty levels of problems in online judge systems," *ACM Trans. Inf. Syst.*, vol. 36, no. 3, pp. 27:1–27:33, 2018.

[6] J. Vie, "Deep factorization machines for knowledge tracing," in *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications@NAACL-HLT 2018, New Orleans, LA, USA, June 5, 2018*, J. R. Tetreault, J. Burstein, E. Kochmar, C. Leacock, and H. Yannakoudakis, Eds. Association for Computational Linguistics, 2018, pp. 370–373.

[7] Z. A. Pardos and N. T. Heffernan, "KT-IDEM: introducing item difficulty to the knowledge tracing model," in *User Modeling, Adaption and Personalization - 19th International Conference, UMAP 2011, Girona, Spain, July 11-15, 2011. Proceedings*, ser. Lecture Notes in Computer Science, J. A. Konstan, R. Conejo, J. Marzo, and N. Oliver, Eds., vol. 6787. Springer, 2011, pp. 243–254.

[8] C. Yeung and D. Yeung, "Addressing two problems in deep knowledge tracing via prediction-consistent regularization," in *Proceedings of the Fifth Annual ACM Conference on Learning at Scale, London, UK, June 26-28, 2018*, R. Luckin, S. Klemmer, and K. R. Koedinger, Eds. ACM, 2018, pp. 5:1–5:10.

[9] 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, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, J. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, Eds. ACM, 2020, pp. 1857–1860.

[10] S. P. Reise, *Item Response Theory*. SAGE Publications, 2016.

[11] J. D. L. Torre, "The generalized dina model framework," *Psychometrika*, vol. 76, no. 2, pp. 179–199, 2011.

[12] J. Zhang, X. Shi, I. King, and D. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, R. Barrett, R. Cummings, E. Agichtein, and E. Gabrilovich, Eds. ACM, 2017, pp. 765–774.

[13] 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 Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 100–115, 2021.

[14] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 1243–1252.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, Eds., 2017, pp. 5998–6008.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778.

[17] R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, "On layer normalization in the transformer architecture," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 10 524–10 533.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015.

[19] M. Feng, N. T. Heffernan, and K. R. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses," *User Model. User Adapt. Interact.*, vol. 19, no. 3, pp. 243–266, 2009.

[20] X. Xiong, S. Zhao, E. V. Inwegen, and J. Beck, "Going deeper with deep knowledge tracing," in *Proceedings of the 9th International Conference on Educational Data Mining, EDM 2016, Raleigh, North Carolina, USA, June 29 - July 2, 2016*, T. Barnes, M. Chi, and M. Feng, Eds. International Educational Data Mining Society (IEDMS), 2016, pp. 545–550.