Crafting Global Information in Mini-batches for Knowledge Tracing

Hui Zhao[1], Yanze Wang[1], and Jun Sun[1]✉

[1]Wangxuan Institute of Computer Technology, Peking University, No. 128 Zhongguancun North Street, Haidian District, Beijing, 100871, China
{hui.zhao, 2201111749}@stu.pku.edu.cn; sunjun@pku.edu.cn

Abstract. Knowledge tracking makes predictions regarding students' future learning performance by leveraging their past answer records. These predictions rely on both the historical answer data and the connections between knowledge concepts. In deep learning, its mini-batch input and end-to-end learning method risks ignoring global information like exercise difficulty, student status, and exercise correlation. Moreover, the form of input data makes predictions uninterpretable. To address these issues, we developed a framework to add crafted global information within mini-batches for the network to learn integrated insights. We designed exercise difficulty index, student confidence index, and exercise relationships as global information. Experimental findings demonstrate that the network enhanced with the added global information can markedly improve the performance of the model. On the knowledge tracing dataset, merely by incorporating global information into each batch, we achieved an average increase of over 3.2% in the AUC metric. Simultaneously, the designed artificially global information also boosts the model's interpretability.

Keywords: Knowledge tracing · Mini-batch input · Global information
.

1 Introduction

Education has long been recognized as a crucial cornerstone of societal development and individual growth [4]. However, traditional one-size-fits-all approaches to teaching have often fallen short in catering to the unique learning requirements of each student. In recent years, online learning systems have broken the barriers of time and space, providing convenience to learners while also accumulating a vast amount of learning interaction data. Online personalized education, also known as individualized or adaptive learning, is an educational paradigm that tailors the learning process to match the distinct needs, abilities, and preferences of each student [5]. The premise of online personalized education is to understand the learning situation of each student. Only by having a clear understanding of a student's cognitive level and learning ability can we provide tailored instruction.

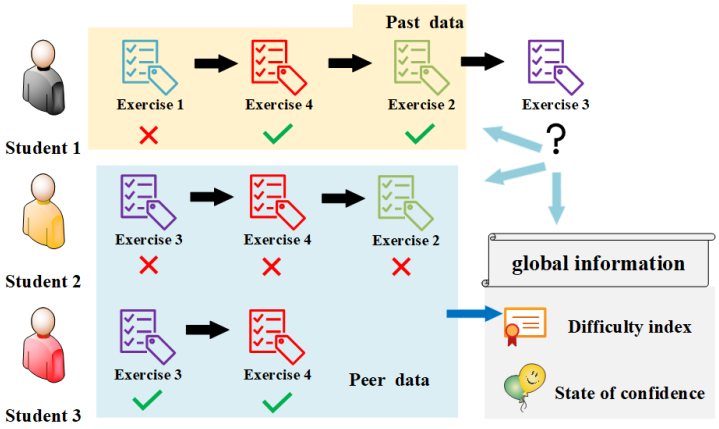

Fig. 1. Knowledge tracking prediction mechanism with global information.

Knowledge tracking (KT) is the task of constructing students' knowledge structures according to the data of students' already answered exercises, and then predicting students' performance on new exercises that corresponding to new knowledge concepts. The performance is measured in terms of probabilities that the students will be able to answer the exercises correctly. KT can continuously predict students' answering performance based on their answering records [18]. Figure 1 illustrates the process of KT prediction. Student 1 answered Exercise 1, Exercise 4, and Exercise 2 respectively. This student answered Exercise 1 incorrectly and the other two exercises correctly. The task of KT is to predict the performance of Student 1 on Exercise 3, that is, the probability that this student can correctly answer Exercise 3. The prediction is based on two aspects. On the one hand, it comes from Student 1's past answering records. On the other hand, it is derived from the data of other students. The data of other students (peer data) contains the relationships between exercises or knowledge concepts, which serves as an important basis for KT prediction [10]. In the KT dataset, the label 0 is used to indicate that a student answers an exercise incorrectly, and the label 1 is used to indicate that a student answers an exercise correctly. The loss function for KT task is to calculate the cross-entropy loss between the predicted values and the true values, and the AUC (Area Under the Curve) metric is adopted as the evaluation indicator [17].

Generally speaking, the input of deep learning often takes the form of mini-batch input. This data input method poses the risk of catastrophic forgetting, which further causes the network to ignore global information such as the relationships between knowledge concepts and the difficulty levels of knowledge concepts.

While shuffling has been effective in mitigating catastrophic forgetting, this approach is primarily suited to labeled data scenarios. Unlike labeled tasks, tasks without labels lack the concept of classes, making it inconvenient to directly add

data from different classes within mini-batches. Moreover, for regression tasks, time series tasks, and self-supervised tasks, catastrophic forgetting manifests in different forms. To address this widespread issue, we design the method which aims to explicitly extract the global information so that the network can learn useful knowledge or patterns in the peer data.

As indicated above, KT fulfills the conditions of being a non-classification, regression-based, and time series-oriented task. In KT, while incorporating information through mini-batch data input, the neural network faces challenges in effectively capturing information. On this account, we design an artificial method to extract information and integrate it into the mini-batches. During the process of informative construction, two kinds of information are input into the network. Specifically, these are the degree of difficulty of the exercises and the students' self-confidence when answering those exercises, which can be seen as Figure 1. The relationship network between knowledge concepts has also been sent in the form of mini-batch input. These two types of information along with the relationship network are referred to as global information. The outcomes of the experiments clearly show that the extra global information we incorporated is capable of boosting the performance of the model. As a result, there is a notable improvement of 3.2% in the AUC metric. Additionally, we carried out qualitative case analyses to illustrate that these two categories of information possess the potential to enhance the interpretability of the model. Our main contributions are presented as follows:

- The performance of the KT model has been enhanced by adding global information.
- A mode of incorporating global information into mini-batches has been designed, which provides valuable references.
- The interpretability of the KT task has been strengthened through the added information.

2 Related Work

2.1 KT task

KT predicts the probabilities that the students will be able to answer the new exercises correctly. In neural networks, KT is designed to be an end-to-end approach, which can directly predict a student's performance on new exercises without the need to obtain the intermediate state of the student's knowledge structure [13]. Indeed, the end-to-end approach may sacrifice interpretability in certain cases [8]. To this end, in this paper, we have taken a different approach by incorporating key global information from cognitive psychology, such as exercise difficulty and student confidence, into the learning process.

In recent times, the pursuit of augmenting the efficacy of KT models has been a focal point for researchers. To this end, they have been integrating diverse learning factors into the frameworks of these models [3]. These factors

span a broad gamut of elements that play a role in students' learning trajectories, with examples being cognitive load [8], motivation [7], and engagement [6]. The integration of these factors into knowledge tracing models provides educators and educational systems with a dual - perspective understanding. It allows them to not only gauge the knowledge students have acquired but also to fathom the learning strategies they employ. Numerous approaches have emerged, signifying the burgeoning efforts to construct all - encompassing models that accommodate the multifaceted aspects of learning. For example, the utilization of neural networks to discern and map out intricate learning patterns [10], or the amalgamation of cognitive models with affective computing techniques [9]. Among these, the Context - Aware Attentive framework [16] stands out. This framework capitalizes on the potency of attention mechanisms while factoring in contextual information to fortify the predictive prowess of the KT model. By incorporating elements like temporal dependencies, student responses, and the sequence of exercises, this approach enables a more profound and nuanced comprehension of the ebb and flow of students' learning dynamics [19].

2.2 Integration of global information

The integration of global information has garnered significant attention in diverse fields, extending beyond KT in educational field. In computer vision, for instance, global context has proven crucial in image understanding tasks. Models like the Global Context Network [11] employ global information to refine object recognition by capturing long-range dependencies between image regions. This approach harnesses global context to enhance the accuracy of object detection and segmentation. In the realm of natural language processing, global information has been harnessed for various applications. The Transformer architecture [12], which underpins models like BERT and GPT, exploits global context through self-attention mechanisms to capture relationships between words in a sentence. This technique facilitates improved language understanding, semantic analysis, and machine translation.

However, whether in KT or other fields that leverage global information, the focus has largely been on the model's perspective. These methods either adjust network parameters through mini-batch learning during model training or employ intricate structures to retain information. Different from previous research methods, we have directly integrated global information into the raw data using specific algorithms. This approach aims to reduce the complexity of the model learning these features directly. Experimental results demonstrate the effectiveness of the type of global information and the added information contributes to enhancing model performance. Moreover, as the introduced global information aligns with cognitive psychology and learning principles, the model also lends itself to strong interpretability.

2.3 Catastrophic forgetting

In deep learning, the network optimizes its parameters through the input of mini-batch data [20]. When the network is trained, all data is traversed for each epoch. In a network with a fixed structure, parameters are the sole variable that encodes dynamic knowledge, and they are adjusted through the input of mini-batches of data. This approach estimates the gradient of the entire dataset using the gradient of the mini-batch data [21]. After multiple rounds of optimization, the network achieves a locally optimal state. Utilizing mini-batches for input into the network and adjusting parameters in this manner overcomes memory limitations, enhances computational efficiency, promotes stable training, and helps to prevent overfitting. Although adjusting parameters with mini-batch input is a common paradigm in deep learning, it has drawback of "catastrophic forgetting."

Catastrophic forgetting refers to the phenomenon where a network forgets previous knowledge after learning new information [20]. To mitigate this issue, researchers have primarily proposed four solutions: Self-refreshing Memory Approaches [22], knowledge distillation method [23], Transfer Technique [24], and Add sample method. The last solution is the most direct way, but it is less applicable to category-free tasks, such as KT. To address the catastrophic forgetting issue in tasks with such data characteristics, we introduce global information to counteract the phenomenon.

3 Method

3.1 Global information

To understand the global information, we consider a dataset $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), ..., (x^{(N)}, y^{(N)})\}$, where $x^{(i)}$ is the feature vector of the i-th sample, and $y^{(i)}$ is the corresponding true label. The total number of samples in the dataset is N. The model is represented by a function $f(\theta, x)$, where θ are the model parameters, and x is the input sample.During training, the data is divided into batches of size B. Each batch is denoted as $D_b = \{(x^{(i)}, y^{(i)})\}_{i \in I_b}$, where I_b is the set of indices for the samples in the batch. The model parameters are updated using the gradient:

$$\theta' = \theta - \alpha \nabla_\theta L_b(\theta)$$

where α is the learning rate and $L_b(\theta)$ is the Loss function. We define the update process as Θ. The parameters of a neural network are updated on mini-batches, and hence the update process in one epoch can be expressed as:

$$P(\Theta_{n+1}^{D_{n+1}} | \Theta_n^{D_n}, \Theta_{n-1}^{D_{n-1}} ..., \Theta_1^{D_1}) = P(\Theta_{n+1}^{D_{n+1}} | \Theta_n^{D_n}), \tag{1}$$

and we expect $P(\Theta_{n+1}^{D_n} | \Theta_n^{D_{n+1}}) = P(\Theta_{n+1}^{D_{n+1}} | \Theta_n^{D_n})$ when shuffle method is used to mix data and the training becomes stable. In other words, to ensure stable training results, the parameter updates between step n and step $n + 1$ need to

be small. This means that the randomness of data shuffling must be sufficiently strong to counteract the diversity of the data distribution. However, in general, the randomness of data shuffling is influenced by the size of the mini-batches, leaving room for improvement. We introduce global information to extract features and enhance the randomness of the data in another form.

3.2 Formulation of KT

From the perspective of cognitive psychology, we are going to conduct an analysis to figure out which data can be regarded as global information. After that, we will devise methods for the extraction of this global information. Consider a learning context where students interact with a series of exercises to acquire knowledge.

Let S denote the collection of all students, E denote the collection of all exercises, and K denote the collection of all knowledge concepts. For every student $s \in S$, there is an interaction with a sequence of exercises $e_1, e_2, \ldots, e_T$, where T represents the length of this sequential arrangement. For each individual exercise e_i within this sequence, the student generates a binary response $r_i \in \{0, 1\}$. Here, $r_i = 1$ indicates that the student has answered the exercise correctly, while $r_i = 0$ implies an incorrect answer. Our objective is to construct a model for the probability that a student will accurately answer the subsequent exercise in the sequence. This model should be based on the student's previous interactions with the exercises and the associated knowledge concepts. Let $K_{s,i}$ be the set of knowledge concepts that are pertinent to exercise e_i for the specific student s. The knowledge state of the student, denoted by KS_s, is represented as:

$$KS_s = f(\cup_{i=1}^{T} K_{s,i}, \cup_{i=1}^{T} r_i, \phi), \tag{2}$$

where the function f serves as the knowledge state function and ϕ stands for the relationship function between knowledge concept. The probability $P(r_{T+1} = 1|g(KS_s), k_{T+1})$ that a student can correctly answer the next exercise can be modeled using a function g. This function g takes into account the student's knowledge state KS_s. By substituting the expression for KS_s from formula (2), we can rewrite this probability as:

$$P(r_{T+1} = 1|g(f(\cup_{i=1}^{T} K_{s,i}, \cup_{i=1}^{T} r_i, \phi), k_{T+1})). \tag{3}$$

In the context of the neural network, both the function f and the function g are trainable. The function f captures the relationships and interconnections among different knowledge concepts and can be learned from the data of other students (peer data). The function g maps the student's state of knowledge mastery to the likelihood of correctly answering a particular exercise. In previous research endeavors, the optimization of the function f was accomplished through the process of fine-tuning with mini-batch data inputs. Nevertheless, this approach may not enable the neural network to effectively learn important features such as the difficulty level of the exercises. Similarly, in prior works, the function g solely predicted the answer states based on the knowledge mastery

state KS_s, completely overlooking the state of the student during the answering process. Even though previous studies adopted an end-to-end approach, which did not fully emphasize the distinct roles of the functions f and g, these oversights can manifest in the input data, ultimately having an impact on the performance of the model. Our proposed approach involves the integration of global information into the functions f and g. More precisely, this entails incorporating information about the difficulty of the exercises and the state of the student's response into these functions. Let d_i represent the difficulty index of exercise e_i, and c_T represent the confidence level of the student in solving the exercise (which is a manifestation of the response state) during the T-th interaction. Then, the student's knowledge state and the probability of predicting a correct answer for the next exercise can be described as follows:

$$KS_s = f(\cup_{i=1}^{T} K_{s,i}, \cup_{i=1}^{T} r_i, \phi, \cup_{i=1}^{T} d_i), \tag{4}$$

$$P(r_{T+1} = 1 | g(f(\cup_{i=1}^{T} K_{s,i}, \cup_{i=1}^{T} r_i, \phi), k_{T+1}, c_{T+1})). \tag{5}$$

The subsequent sections will detail the design of the computational methods for d_i and c_T.

3.3 Algorithm

Exercise difficulty index. The global information, encompassing both the exercise difficulty index and the confidence index, is derived from the original dataset. In order to compute these two types of information, we define a student's learning record as $LR_n = (lr_1, lr_2, \ldots, lr_n)$, where lr_i corresponds to the i-th interaction within the learning process. Let $\Psi = (LR_n, LQ_m, \ldots)$ denote the collection of all students' learning interactions. Similar to LR_n, LQ_m represents the learning record of another student. For each individual element lr_i in the learning record, we have $lr_i = (k_i, r_i)$, where k_i is a knowledge concept belonging to the set K, and r_i is the response label associated with the knowledge concept k_i. The value of r_i is a binary quantity, taking on either the value of 0 or 1, indicating an incorrect or correct answer respectively. For every knowledge concept k in the set K, we define its difficulty index d_k. The calculation of d_k is carried out as follows:

$$d_k = \frac{\sum_{A \in \Psi} (\sum_{i=1}^{|A|} count(k_i = k, r_i = 0))}{\sum_{A \in \Psi} (\sum_{i=1}^{|A|} count(k_i = k))} \times SA, \tag{6}$$

Here, the function *count* is used to determine the number of occurrences of the conditions specified within the parentheses, and SA (student ability) represents a matrix whose values can be learned in the network.

Confidence index. We introduce the concept of the confidence index, denoted as c, which represents a student's state after finishing a specific exercise. Its value lies in the range between 0 and 1. We further define the initial states

$c_{init} = 0.5$, which represent the initial confidence index value. Moreover, let decay denote the decay index, which is a real number between 0 and 1. The calculation formula for c in each learning interaction is described as follows:

$$c_i = \begin{cases} c_{init}, & \text{if } i = 1 \\ \frac{0+c_{i-1}}{2} \times \text{decay}, & \text{if } i > 1 \text{ and } r_i = 0 \\ \frac{1+c_{i-1}}{2} \times \text{decay}, & \text{if } i > 1 \text{ and } r_i = 1 \end{cases} \tag{7}$$

From formula (6) and formula (7), it is obvious that both d_k and c_i are bounded between 0 and 1. To improve the differentiation between these two variables and make the embedding operations easier, we carried out a process of ordinal scaling on them. In practical implementation, the two variables were divided by 0.1 and then rounded, resulting in ten discrete levels ranging from 1 to 10. The subsequent sections will elaborate on the utilization of the processed data.

The network of integrated knowledge concept relationships. To incorporate the relationships between knowledge concepts into the network, we introduce the static graph neural network. The knowledge concepts are regarded as nodes and the relationships between knowledge concepts are regarded as edges. When a certain student answers two exercises correctly in an answer sequence, it is considered that these knowledge concepts linked with the two exercises are related (use matrix R_{kc} to represent the relationship); when the student answer two exercises wrongly in an answer sequence, it is also considered that these knowledge concepts are related (use matrix W_{kc} to represent the relationship).

Given $G = (V, E)$, in which $V = \{kc_1, kc_2, ..., kc_n\}$ and kc is the certain knowledge concept. $E \subseteq V \times V$ is the set of edges. We define R_{kc} or W_{kc} as adjacent matrix, and we iterate over all the record on the dataset. If the answer statuses of the i-th and j-th kc are the same in a certain record, then $R_{kc}(i, j) = R_{kc}(i, j) + 1$ or $W_{kc}(i, j) = W_{kc}(i, j) + 1$. Suppose D_{rkc} and D_{wkc} are the degree matrix for R_{kc} and W_{kc}, then $L_{rkc} = D_{rkc} - R_{kc}$ and $L_{wkc} = D_{wkc} - W_{kc}$ are the Laplacian matrix.

$$\frac{\partial u(t)}{\partial t} = -Lu(t),$$

where $u(t)$ is a function that represents the values on the vertices changing with time t. Suppose the eigenvalues of L (L_{rkc} or L_{wkc}) are $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ and the corresponding eigenvectors are $\varphi_1, \varphi_2, \cdots, \varphi_n$. Then the solution of the equation can be expressed as:

$$u(t) = \sum_{k=1}^{n} c_k e^{-\lambda_k t} \varphi_k,$$

where c_k are coefficients determined by the initial conditions. In the deep neural network, we make the Laplacian matrix change only based on the values of each mini-batch input and not based on the backpropagation algorithm. The Laplacian matrix is followed by a fully connected network, and the parameters of this fully connected network change with the backpropagation algorithm.

3.4 KT model

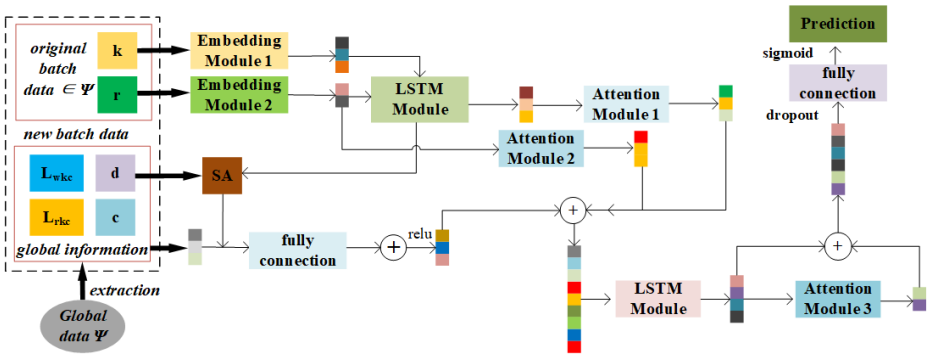

Fig. 2. The KT Backbone network. In the diagram, k, r, d, c represents the knowledge concept index, response lable, difficulty index and confidence index.

As can be seen from Figure 2, the KT model is composed of embedding modules, attention modules, an LSTM (Long Short-Term Memory) module, and fully connected modules. In order to preserve the original information to the greatest extent, we adopt the method of "skip connection" between each module. In the figure, the small colored squares represent the vectors processed by the module. We focus on improving the proposed model from the perspective of data input. Previous methods only input the original batch information, while our method extracts the global information data from the global data and adds it to the newly defined batch input data in the form of four types of data. Among these four types of data, the information about the difficulty of the exercises and the information about students' confidence are what we need to explore. In Figure 2, SA is a fully connected layer, which corresponds to Formula 6. It reflects the ability levels of different students and adjusts its own parameters through the students' answering records.

We added the Laplacian matrix of the relationships between knowledge concepts in each batch input. The operator of this matrix is directly connected to the fully connected network. The matrix is updated whenever each batch is input, and it is not affected by the backpropagation algorithm.

The embedding module plays a crucial role in integrating information by transforming the three continuous variables within the dataset into discrete, lower-dimensional representations. The LSTM module excels at recognizing long-range dependencies, which parallels the knowledge dependency present when students do exercises. Hence, it has been integrated into the network. The attention module adopts a traditional attention mechanism, enabling it to learn which information within the representations is more significant. The reason for not incorporating an attention module in the embedded representations of variables d and c is that these two types of information have already been manually

extracted and processed. They respectively represent global information within the sequence.

4 Experiments

4.1 Datasets

Table 1. The details of the four benchmark datasets

Dataset	Students	Concepts	Interactions
ASSISTments2009	4151	110	325,637
Statics2011	333	1,223	189,297
ASSISTments2015	19,840	100	683,801
ASSISTments2017	1,709	102	942,816

We conducted experiments on four datasets respectively. The attribute values of these four datasets are shown in detail in Table 1. It is important to note that different exercises may share the same knowledge concept, and a single knowledge concept might correspond to multiple different exercises. In a real dataset, each student's learning interaction record is represented as a data entry. These interaction data include identifiers for the corresponding exercises and the student's success or failure in answering them.

Details of the experiment in the narrative version are in Appendix A.2.

4.2 Comparison with other methods

Table 2. The performance of different methods on the AUC scale.

Method	Assistments2009	Statics2011	Assistments2015	Assistments2017	Average
BKT+	≈0.69	≈0.75	-	-	-
DKT	0.8170±0.0043	0.8233±0.0039	0.7310±0.0018	0.7263±0.0054	≈0.7744
DKT+	0.8024±0.0045	0.8301±0.0039	0.7313±0.0018	0.7124±0.0041	≈0.7691
AKT	0.8169±0.0045	0.8265±0.0049	0.7828±0.0019	0.7282±0.0037	≈0.7886
SAKT	0.7520±0.0040	0.8029±0.0032	0.7212±0.0020	0.6569±0.0027	≈0.7333
DKVMN	0.8093±0.0044	0.8195±0.0041	0.7276±0.0017	0.7073±0.0044	≈0.7659
MCB	0.8059	0.8130	-	0.7141	-
ATKT	0.8244±0.0032	0.8325±0.0043	0.8045±0.0097	0.7297±0.0051	≈0.7978
CGIMKT	**0.8546±0.0019**	**0.8381±0.0037**	**0.8783±0.0028**	**0.7495±0.0032**	**≈0.8301**

We compared the performance of different methods ([17], [13], [14], [1], [15], [16], [25], [2]) on the AUC (Area Under the Curve) scale. From Table 2, we can

find that our method outperforms the other methods. Specifically, our method exhibits an improvement of 3.02% and 7.38% over the second-best method on the datasets of Assistments2009 and Assistments2015, respectively. However, on the other two datasets, the improvement over the second-best method is less than 1% and 2%. Upon analysis, we found that the average numbers of sequence used for training in these four datasets are 2500, 230, 11904, and 1180, respectively. From this aspect, we found the larger the dataset, the more pronounced the improvement. This observation suggests that the structure of our network is more suitable for large datasets than other networks, because the incorporation of global information could be more effective in case of low-level shuffle for bigger data. The model might struggle to learn global patterns in the traditional paradigm of adjusting parameters based on mini-batch inputs. Through the extraction and embedding of global data, and by directly feeding these data into the model, we can effectively assist the model in learning the characteristics of the data.

4.3 Ablation study

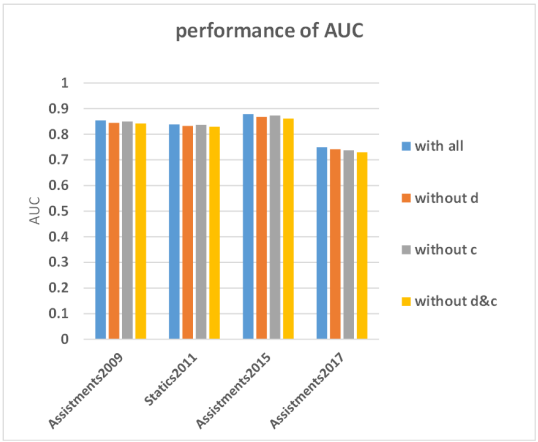

Fig. 3. Ablation experiments with or without difficulty index and confidence index.

We conducted three additional sets of experiments. In the first set, we removed the global information of exercise difficulty data. In the second set, we removed the hidden information of confidence index data. In the third set, both types of the information were removed. As shown in Figure 3, these ablation experiments demonstrated decreased performance compared to the original experiment. The extent of performance degradation varied with the removal of different types of information. From Figure 3, it is evident that removing the confidence index data resulted in a slight decrease in model performance. This

indicates that the model can partially learn from such confidence information. In contrast, removing the difficulty information led to a significant decrease in model performance, suggesting that the model has difficulty learning from this global information. The inclusion of these two types of information provides the model with additional data, naturally optimizing the model's parameters.

More ablation experiments are shown in Appendix A.1.

4.4 Qualitative and interpretability experiments

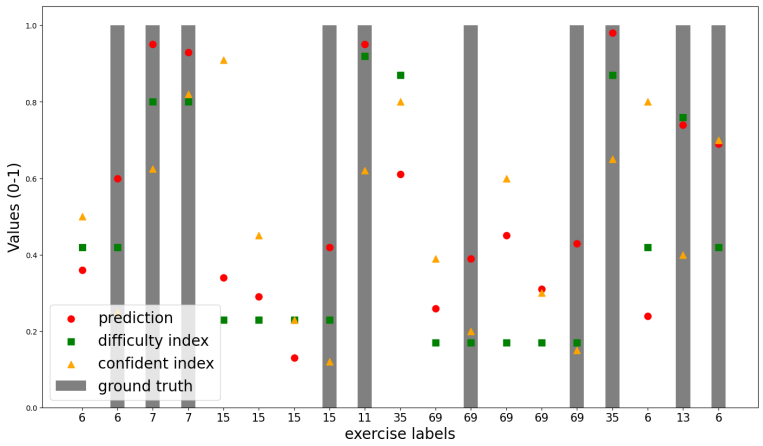

Fig. 4. Schematic diagram of the qualitative experiment on global information.

We explored the influence of two types of global information on the prediction results and conducted a qualitative study on KT task of a certain student. As shown in Figure 4, the horizontal axis is composed of a series of knowledge concept sequences. A student may have multiple interactions with a certain knowledge concept. The vertical axis represents the true values or predicted values. We have also included the two types of global information in the figure as real numbers between 0 and 1. The gray columns indicate that the student has correctly answered the questions in reality, while without the gray column indicates incorrect answer. The predicted values of the KT model are represented by red dots. The difficulty index of the exercises is represented by green squares, and the confidence index of answering the exercises is represented by orange triangles.

As can be seen from Figure 4, the vast majority of the predicted values are close to the real data, indicating that the KT model has good performance. Judging from the difficulty index of the exercises, it is relatively close to the predicted

values, which to a certain extent reflects that the difficulty of the exercises is the main determinant of the prediction. It should be noted here that, due to the addition of the SA module in the model, it can give different exercise difficulty values according to the answering records of different students. The confidence index of students reflects their state during the process of answering exercises, and it is also an important basis for the interpretability of the KT model. In Formula 6, the defined confidence index lags behind the actual answering process, so it contributes less to the interpretability of the model. However, the method proposed in this paper only provides an important reference for the interpretability of the model, and there is no exact corresponding relationship. For example, when the label numbered "35" appears for the first time, the predicted value, the exercise difficulty index, and the confidence index are all high, but in fact, the student answered this exercise incorrectly.

5 Conclusion

In order to tackle the issues of catastrophic forgetting and the neural network's incapability of learning global patterns from mini-batch input data, we incorporate artificially engineered global information into the KT network for verification purposes. This additional information not only boosts the model's performance but also makes the KT predictions more interpretable. In terms of experiments, the approach we put forward surpasses previous methods, attaining the optimal results on every dataset. The qualitative experimental cases we devised offer an illustration of the connection between the two types of introduced information and the predicted results. Nonetheless, the method presented in this paper has its drawbacks. It is highly dependent on domain knowledge. Only when there is a certain degree of comprehension of the task can an efficient extraction plan for global information be formulated.

References

1. Yeung, C. K., & Yeung, D. Y. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In Proceedings of the Fifth Annual ACM Conference on Learning at Scale,1-10.
2. Guo, X., Huang, Z., Gao, J., Shang, M., Shu, M., & Sun, J. 2021. Enhancing Knowledge Tracing via Adversarial Training. In Proceedings of the 29th ACM International Conference on Multimedia, 367-375.
3. Liu, Q., Shen, S., Huang, Z., Chen, E., & Zheng, Y. 2021. A survey of knowledge tracing. arXiv:2105.15106.
4. Chabbott, C. and Ramirez, F.O. 2000. Development and education. In Handbook of the Sociology of Education, 163-187.
5. Keppell, M. 2014. Personalised learning strategies for higher education. In The future of learning and teaching in next generation learning spaces, 3-21.
6. Mongkhonvanit, K., Kanopka, K. and Lang, D. 2019. Deep knowledge tracing and engagement with moocs. In Proceedings of the 9th international conference on learning analytics & knowledge, 340-342.

7. Liu, Z., Chen, J. and Luo, W. 2023. Recent Advances on Deep Learning based Knowledge Tracing. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, 1295-1296.
8. Skulmowski, A. and Xu, K.M. 2021. Understanding cognitive load in digital and online learning: A new perspective on extraneous cognitive load. Educational psychology review, 1-26.
9. Conati, C., Jaques, N. and Muir, M. 2013. Understanding attention to adaptive hints in educational games: an eye-tracking study. International Journal of Artificial Intelligence in Education, 136-161.
10. Liu, D., Zhang, Y., Zhang, J., Li, Q., Zhang, C. and Yin, Y.U. 2020. Multiple features fusion attention mechanism enhanced deep knowledge tracing for student performance prediction. IEEE Access, 194894-194903.
11. Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. 2017. Large kernel matters—improve semantic segmentation by global convolutional network. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4353-4361.
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. 2017. Attention is all you need. Advances in neural information processing systems, 30.
13. Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L. J., & Sohl-Dickstein. 2015. Deep knowledge tracing. Advances in neural information processing systems, 28.
14. Zhang, J., Shi, X., King, I., & Yeung, D. Y. 2017. Dynamic key-value memory networks for knowledge tracing. In Proceedings of the 26th international conference on World Wide Web, 765-774.
15. Pandey, S., & Karypis, G. 2019. A self-attentive model for knowledge tracing. arXiv:1907.06837.
16. Ghosh, A., Heffernan, N., & Lan, A. S. 2020. Context-aware attentive knowledge tracing. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, 2330-2339.
17. Yudelson, M. V., Koedinger, K. R., & Gordon, G. J. 2013. Individualized bayesian knowledge tracing models. In International conference on artificial intelligence in education, 171-180.
18. Abdelrahman, G., Wang, Q. and Nunes, B., 2023. Knowledge tracing: A survey. ACM Computing Surveys, 55(11), pp.1-37.
19. Shen, S., Liu, Q., Huang, Z., Zheng, Y., Yin, M., Wang, M. and Chen, E., 2024. A survey of knowledge tracing: Models, variants, and applications. IEEE Transactions on Learning Technologies.
20. LeCun, Y., Bengio, Y. and Hinton, G. Deep learning. nature, 521(7553), pp.436-444, 2015.
21. McCandlish, S., Kaplan, J., Amodei, D. and Team, O.D. An empirical model of large-batch training. arXiv preprint arXiv:1812.06162, 2018.
22. Shmelkov, K., Schmid, C. and Alahari, K. Incremental learning of object detectors without catastrophic forgetting. In Proceedings of the IEEE international conference on computer vision, 2017.
23. Li, Z. and Hoiem, D. Learning without forgetting. IEEE transactions on pattern analysis and machine intelligence, 40(12), pp.2935-2947, 2017.
24. Lee, S.W., Kim, J.H., Jun, J., Ha, J.W. and Zhang, B.T. Overcoming catastrophic forgetting by incremental moment matching. Advances in neural information processing systems, 2017.

25. Lee, U., Park, Y., Kim, Y., Choi, S. and Kim, H., 2024, June. Monacobert: Monotonic attention based convbert for knowledge tracing. In International Conference on Intelligent Tutoring Systems (pp. 107-123). Cham: Springer Nature Switzerland.

A Appendix

A.1 Ablation study of modules in the model

Table 3. Ablation experiments on individual components of the model. Other combinations of modules were not included in the table as they lack real network topologies.

Attention	LSTM	Concatenation	AUC (average)
√	√	-	82.11%
√	-	√	77.05%
√	-	-	76.78%
-	√	-	79.13%
√	√	√	82.59%

We explored the importance of each module in improving model performance. Throughout the multi-round experiments, we systematically removed different modules to observe the contribution of each module to the model's performance. In Table 3, only use the attention module, the AUC value achieved 76.78%, while the AUC value reach to 79.13% only use the LSTM module. We observed that the LSTM module has the most significant impact on the model's performance, making it a crucial component for network. LSTM, as a variant of recurrent neural networks, excels at capturing long-term dependencies. In the context of the KT task, the knowledge concepts scattered across the data representations are effectively learned by the LSTM module. Intuitively, there exist relationships between different exercises when students solving exercises, which can be learned by the neural network. The attention modules enhance the model's performance on the basis of LSTM module, but the standalone impact is not as substantial as LSTM module. We refer to the catenate mechanism applied to Attention Module 3 as the concatenation trick. Compared to the previous two modules, the concatenation trick provides a moderate enhancement to the model's performance, yet it retains more original information within the network. However, it's important to note that the vector embedding module is also highly essential. Isolating the vector em-bedding module for separate comparison might not be suitable since embedding is a fundamental data-handling operation employed by previous methods. The two types of additional information are both subject to processing through embedding layers before being integrated. Each of the modules mentioned above has a positive impact on enhancing the model. And we

observed that the modules prove to be more effective in enhancing the model's performance when they are cleverly combined.

A.2 Implementation details

For parameter selection, we have taken into thorough consideration the uniqueness of each dataset. Concerning the embedding of knowledge concepts, we have accounted for the number of knowledge concepts and the total count of interactions in the dataset. Generally, as these two values increase, a higher embedding dimension is set to enable a more detailed representation of the relationship of the knowledge concepts and learning interactions. The knowledge concept embedding dimensions we have set for these four datasets are 256, 512, 64, and 256, respectively. Concerning the embedding of learning interaction response, we have considered the number of students and the total count of interactions in the dataset. A higher embedding dimension is preferred when these values are larger, as this ensures a more robust representation vector for capturing the relationships between students and the mastery degree of exercises. The learning interaction response embedding dimensions we have set for these four datasets are 80, 60, 30, and 60, respectively. Since in the KT network, the learning interaction response embedding vector is summed with the global information embedding vector and the hidden in-formation embedding vector, the dimensions of the latter two vectors are kept consistent with the former.

Our method is trained end-to-end by minimizing the bi-nary cross-entropy loss of all learner responses to optimize the parameters in the network. We employed the Adam optimizer with a learning rate of 0.001 and incorporated a patience strategy set to15. The dimensions of several attention modules were uniformly set to 120. Due to the introduction of additional features and multiple concatenations within the network architecture, the risk of overfitting was considered. As a mitigation measure, we introduced dropout layers at the end of the network. The dropout values were chosen as 0.5, 0.9, 0.1, and 0.5, respectively. The basis for selecting these values is a consideration of the average lengths of data records across the four datasets. The dimension of the output of the fully connected layer is consistent with the number of knowledge concepts in the datasets. When calculating the confidence index c, we set c_{init} to 0.5 and *decay* to 1. Finally, the hyper-parameters of the other modules can be computed through connected logic.

See the code for more details.