Smart Tutor: Personalized Performance Analysis

Leona E Joseph
lj9984@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Abstract

This work presents Smart Tutor, an intelligent, adaptive learning environment that tracks student progress to predict the student's knowledge at any point. The stimulus arises from the constraints of standard "one-size-fits-all" learning as well as the misuse of AI technologies for academic shortcuts. The task is to model both the temporal aspect of learning and the structural concepts' inter-relational aspects properly. To address this, we bridge Learning Process Knowledge Tracing (LPKT) and Graph Neural Networks (GNN), integrating temporal dynamics and concept interdependencies in one framework. The approach applies the EdNet KT1 dataset using graph-augmented knowledge updates and student interaction sequences. We anticipate that our model will be superior to standard LPKT and GNN baselines, enabling more accurate prediction of student responses and paving the way for highly personalized learning.

1 Introduction

Problem Statement: The problem is creating a personalized learning system using AI to analyze their strengths and weaknesses. The system would identify areas where the students struggle or lack preparation. These areas will be prioritized in the recommendation. The system will suggest tutorials and quizzes with the weaker areas more in mind than the stronger areas. This tailored approach will help students improve their learning by addressing their individual challenges and helping them progress more efficiently. The process is to start small with quizzes and recommendation systems and then with performance and data availability try scaling it up to creating personal tutorial methods if possible.

2 Motivation

The main issue with today's education is that the idea of one model fits all. Each student is unique with different strengths and weaknesses. The idea is to find these differences and cater the learning program for each student. The other issue is the existence of various LLMs that can help students cheat through education without them learning the content. Students reach the end of any query without understanding the process. It is important to use AI to fight AI. The personalized AI helps students understand where they should focus and gives them good tutorials to follow and learn from. The AI will also create engaging quizzes and present the problem with innovative approaches. The data for this problem statement would be the publicly available Kaggle datasets with students' performance in exams and synthetic data created using learning patterns

3 Background and Related Work

Knowledge tracing (KT) is a field that's seen tons of research, with different methods trying to model and predict how students learn. For example, Yang and colleagues came up with a Graph-based Interaction Model for Knowledge Tracing (GIKT). It uses Graph Convolutional Networks (GCN) to try and understand the long-term connections between students and questions. However, it can struggle to scale up and depends heavily on knowing the relationships between questions and skills[1].

To make online learning more personal, Xiao et al. built a recommendation system that uses association rules, content filtering, and collaborative filtering. The downside is that it's hard to scale and has trouble with feature extraction.

Shen et al. introduced Learning Process-consistent Knowledge Tracing (LPKT), which is interesting because it tries to match knowledge states with how people actually learn, using a neural network. But, it uses a pretty basic way of representing knowledge and assumes that time and learning rate are strongly linked, which might not always be true.

To get around some of the problems with Transformer-based KT models, Yang et al. created the Evolutionary Neural Architecture Search for Transformer in Knowledge Tracing (ENAS-KT). They used NAS to fine-tune transformer structures. Even so, it's still quite computationally expensive and doesn't generalize as well as we'd like.

Finally, Long et al. developed Individual Estimation Knowledge Tracing (IEKT), which personalizes predictions by using modules to estimate cognition and acquisition. Unfortunately, it tends to overfit the data and has high computational costs because it uses reinforcement learning.

Overall, these studies really highlight the ongoing efforts to improve KT models by adding advanced techniques, but there are still significant challenges to overcome in areas like scalability, personalization, and just making them efficient to run.

4 Methodology

The proposed Smart Tutor system applies a multi-stage modeling pipeline that is designed to monitor and enhance student learning through sequential and structural modeling. The system begins with student interaction data in terms of question attempts, answers, and timestamps. This is then passed through embedding layers, where exercises (et), concepts (ct), and timestamps (tt) are converted to dense vectors. These representations are then propagated through the Learning Process Knowledge Tracing (LPKT) procedure, which transforms the latent student knowledge state over the long run (hLPKTt) while accounting for learning and forgetting processes. A Graph Neural Network (GNN) layer also extracts relational relationships among concepts and produces a graph-augmented state (hGt). These two states of knowledge are then fused within the joint update module (ht), integrating sequential and relational understanding. The resultant representation is passed to the prediction layer, which estimates the probability of a correct student response (pt). The model is trained using a binary cross-entropy loss function (L), optimized using gradient descent, and validated with respect to its performance prediction accuracy.

4.1 Problem Formulation

Learning Process Knowledge Tracing (LPKT) models a student's evolving knowledge state based on their interaction history. Given a sequence of exercises $X = \{x_1, x_2, ..., x_T\}$, responses $A = \{a_1, a_2, ..., a_T\}$, and timestamps $D = \{d_1, d_2, ..., d_T\}$, the knowledge state at time t is updated as:

$$h_t = f(h_{t-1}, x_t, a_t, d_t) \tag{1}$$

where:

- h_t represents the knowledge state,
- f is a function incorporating learning and forgetting mechanisms.

Graph-Based Knowledge Representation is defined as $G = (V, E)$, where:

- V represents knowledge concepts (nodes),
- E represents relationships between concepts (edges).

Each node $v_i \in V$ has a feature vector $\mathbf{v_i}$, initialized via domain knowledge or data-driven methods.

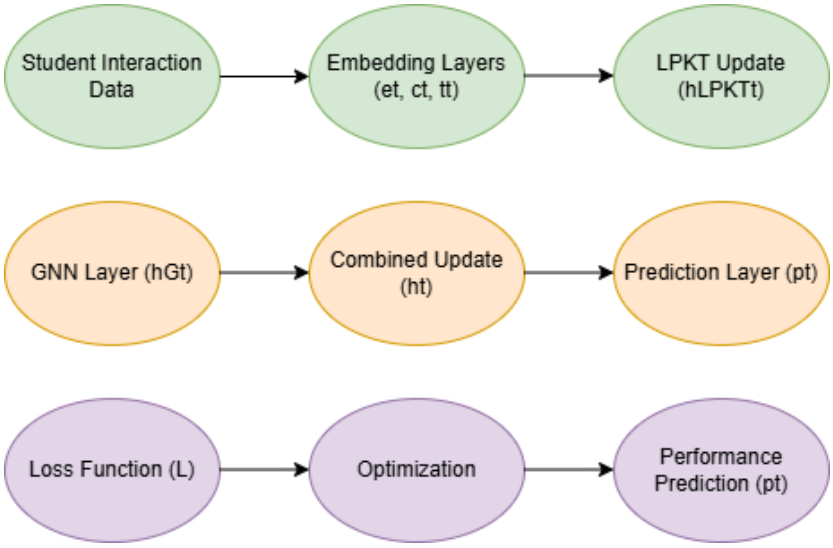

Figure 1: Knowledge Tracing System Workflow

4.2 Model Architecture

The Embedding Layers (LPKT) are:

$$e_t = W_e x_t \tag{2}$$
$$c_t = W_c v_t \tag{3}$$
$$t_t = W_t d_t \tag{4}$$

In the Graph Neural Network (GNN) Layer a message-passing mechanism updates concept embeddings:

$$\mathbf{h}_t^G = \sigma\left(\sum_{(i,j)\in E} \alpha_{ij} W_r \mathbf{h}_j^{t-1}\right) \tag{5}$$

where:

- α_{ij} is an attention weight,
- W_r is a learnable relation matrix,
- σ is an activation function.

4.3 Knowledge State Update Process

The LPKT Knowledge is updated as:

$$h_t^{LPKT} = h_{t-1} - \gamma f_{forget}(h_{t-1}, \Delta t) + f_{learn}(x_t, a_t, d_t) \tag{6}$$

The Graph-Enhanced update is:

$$h_t^G = g(G, h_t^{LPKT}) \tag{7}$$

The Combined Update Rule:

$$h_t = \beta h_t^{LPKT} + (1-\beta) h_t^G \tag{8}$$

where β controls the balance between LPKT and graph-enhanced knowledge states.

4.4 Prediction Layer

The final performance prediction:

$$p_t = \sigma(W_1 h_t + W_2 \mathbf{g}(c_t) + b) \tag{9}$$

where $\mathbf{g}(c_t)$ extracts contextual graph information.

4.5 Training Process

- **Loss Function:** Binary cross-entropy

$$\mathcal{L} = -\sum_{t} (a_t \log p_t + (1 - a_t) \log(1 - p_t)) \tag{10}$$

- **Optimization:** Gradient-based optimization on both LPKT and GNN components.
- **Graph Learning:** Optional updates to graph structure based on performance.

4.6 Comparison with Baseline Approaches

A Standard LPKT has sequential dependencies but lacks structured knowledge representation. Does not explicitly capture relationships between concepts. In a Graph-Based approach models represent concept relationships but ignore individual learning dynamics. They also lack a temporal component for forgetting and reinforcement. The Proposed Approach captures time-sensitive learning and forgetting, a strength of the LPKT approach, and encodes structural knowledge for better generalization, A graph representation strength. So the proposed approach integrates both approaches for a comprehensive knowledge-tracing model. The Concepts are interlinked, preventing isolated learning. The advantages of LPKT's forgetting mechanism are maintained. The model also learns structural relationships, enabling knowledge transfer.

5 Dataset

5.1 Dataset Used

The data set used for this work is EdNet: KT1, a large education data set that records student interactions with an online learning system. It has over 100 million interactions of over 700,000 students engaging with 13,000+ questions on numerous concepts. EdNet KT1 was selected because it has vast and diverse data, recording a wide range of student responses and actions, and thus is most appropriate for modeling learning patterns. It also provides temporal interaction data, with timestamps on every question attempt, which is essential when using the LPKT model to simulate student knowledge development over time. Additionally, in the dataset, concept mapping exists, where question IDs are equated to concepts, allowing for the creation of a knowledge graph that can be used for GNN incorporation. The dataset contains interaction data, such as question attempts, answers (correct/incorrect), and timestamps. It also provides concept and skill mappings, tagging each question with one or more concepts to facilitate knowledge tracing. The exact timestamps log the time of each interaction, which is critical for modeling forgetting and reinforcement in student learning. The EdNet KT1 dataset is publicly available on Kaggle: EdNet: $KT1$.

The EdNet $KT1$ data set contains a few significant columns of student interaction data. The 'timestamp' axis captures the time for each question attempt so that temporal modeling of students' knowledge evolution is possible. The $solving_id$ is a unique ID for each interaction, and the $question_id$ captures the specific question attempted.

The $user_answer$ contains the student's selected answer and the $elapsed_time$ contains the time (in milliseconds) taken by the student to respond. The target variable of this data set would typically be the student's accuracy of response, which may be derived by comparing the $user_answer$ with the correct response and hence predicting student performance and retention of knowledge over time.

5.2 Data Preprocessing

Preprocessing will be conducted on the EdNet KT1 dataset to prepare it for LPKT and GNN modeling. Cleaning of the data will first include removing incomplete responses by removing rows with missing values in question attempts, answers, or timestamps and removing interactions that do not map to a recognized concept for keeping knowledge tracing consistency. Second, data transformation will hold responses as binary labels with the correct response being denoted by 1 and the incorrect response by 0. The timestamps will be normalized by having them as time differences (Δt) between

successive attempts in order to consider the temporal nature while modeling forgetting and reinforcement in LPKT. At feature engineering, question attempt sequences, responses, and timestamps will be formed for each student in order to represent knowledge states using LPKT. Besides, a graph will be constructed such that nodes will represent ideas and edges will signify idea relationship by question similarity or requirement prerequisites for enabling integration of GNN. Finally, a temporal train-test split will be employed to ensure time coherence where 80 percent of the data will be set for training to accommodate student knowledge states and 20 percent reserved for testing for model performance on unobserved interactions. These preprocessing steps will ready the EdNet KT1 dataset for precise modeling of students' learning progression with LPKT and knowledge state refinement with GNN for concept-level interactions.

6 Experimental Design

6.1 Performance Metrics

Accuracy (ACC) – The fraction of correct predictions (i.e., predicted answer matches the student's actual outcome). Accuracy is a straightforward indicator of overall model correctness and is easy to interpret (higher accuracy means the model's predictions align with more actual answers). However, accuracy alone can be misleading if the data are imbalanced, so it is considered alongside more discriminative metrics.

Area Under the ROC Curve (AUC) – AUC measures the model's ability to distinguish between correct and incorrect responses across all classification thresholds. An AUC of 1.0 indicates perfect discrimination, while 0.5 indicates no better than chance. AUC is appropriate here because the model outputs a probability of a student answering correctly; evaluating the ROC curve captures the quality of these probability estimates. It is especially useful if the proportion of correct vs. incorrect responses is imbalanced, as AUC evaluates ranking performance rather than raw accuracy.

Root Mean Square Error (RMSE) – Track RMSE between the predicted probability of success and the actual outcome (0 or 1). This is a regression-oriented metric that captures the average prediction error in terms of probability. A lower RMSE indicates the model's confidence estimates are closer to the true outcomes on average. RMSE complements accuracy and AUC by penalizing overly confident wrong predictions and rewarding well-calibrated probabilities. This provides insight into how well-calibrated and fine-grained the predictions are.

6.2 Experimental Setup

The experiments are designed to rigorously evaluate the model on temporal student-interaction data while avoiding information leakage. Key aspects of the experimental procedure include: Temporal Train-Test Split – Given the chronological nature of student learning data, a temporal hold-out strategy will be employed rather than random shuffling. Specifically, each student's interaction sequence is split so that the earliest 80 percent of interactions serve as the training set and the most recent 20 percent as the test set, preserving time order. This ensures that the model is always trained on past events and tested on future events, mirroring a real-world scenario of predicting upcoming performance. Using a chronological split prevents any future information from leaking into training, thus maintaining the integrity of evaluations on "unseen" future student responses. A small validation set may be further held out from the training portion for hyperparameter tuning and early stopping, but all testing is strictly on future interactions. In this study the primary evaluation uses the single 80/20 temporal split described above, which is a common approach in knowledge tracing to simulate forward-in-time prediction. This approach balances simplicity and fidelity to the temporal learning process.

Multiple Runs for Robustness – To account for any randomness in model initialization or training (e.g., random weight initializations, mini-batch sampling order), each experiment is repeated multiple times. For instance, the model is trained and evaluated $N = 5$ times (with different random seeds) and report the average performance and standard deviation across these runs. In essence, this approach provides a more reliable estimate of true performance by averaging out random fluctuations.

Baseline Comparisons –The baseline models are to be trained (e.g., the standard LPKT model without graph augmentation, and a graph-only model without LPKT's temporal modeling) under the same conditions for fair comparison. They will use the identical train/test split and be subject to the

same preprocessing and training regimen. By evaluating all models on the same temporally segmented test data, it is ensured that the performance differences can be attributed to the modeling approach rather than data differences. If necessary, hyperparameters for each baseline will be tuned on the validation portion of the training set to give each model a fair chance to perform optimally.

6.3 Statistical Significance Testing

To determine whether the proposed integrated model (LPKT + GNN) significantly outperforms the baseline methods, a statistical significance test will be conducted on the evaluation metrics collected across repeated runs (or folds): Paired t-Test (two-tailed) – For each baseline comparison, a paired t-test was used to compare the performance scores of this model vs. the baseline model across the multiple runs or folds. In a paired setup, each run yields a performance metric for the proposed model and the baseline on the same test data split; the difference in these paired scores across runs is tested for a mean significantly different from zero. The paired t-test is appropriate here because it accounts for the fact that results are obtained on the same data splits. A significant level of $\alpha = 0.05$ will be applied. If the p-value from the t-test is below 0.05, it is concluded that the model's improvement is statistically significant (with the null hypothesis being that there is no difference in mean performance). There will also be reported p-values to quantify confidence in this model's gains.

Wilcoxon Signed-Rank Test – In addition to the t-test, a Wilcoxon signed-rank test will be employed as a non-parametric alternative, especially if the distribution of differences deviates from normal or the sample of runs is small. The Wilcoxon test makes no normality assumption, instead using the ranks of differences, and is recommended in the machine learning literature as a robust test for comparing two classifiers on paired measurements. It essentially tests whether the median of the differences is zero. This test is often seen as a backup to the paired t-test when its assumptions are in doubt, to ensure that any observed advantage of this model is not an artifact of statistical assumptions. A significant Wilcoxon test (p ¡ 0.05) would corroborate the t-test results, giving more confidence that the improvement holds in a distribution-free sense.

7 Expected Results

Integrating LPKT + GNN model can give us a model that will outperform both traditional LPKT and standalone graph-based models across key evaluation metrics such as Accuracy, AUC, and RMSE. Specifically, we anticipate at least a 3–5 percent improvement in AUC and a noticeable reduction in RMSE over baseline models. A statistically significant p-value (¡ 0.05) from both the paired t-test and Wilcoxon signed-rank test will confirm the superiority of our model. These improvements will demonstrate the advantage of incorporating both temporal learning behavior and structural knowledge into one unified system.

8 Potential Impact

The proposed model addresses the long-standing issue of "one-size-fits-all" learning through the utilization of AI for analyzing individual learning patterns and providing personalized learning paths to every student. Personalization has the potential to significantly improve learning efficiency by targeting areas of weakness for the students, improve student confidence through demonstration of concrete improvement in personalized metrics, and reduce dropout rates in online learning platforms through maintenance of higher levels of engagement. During a period where LLMs and AI tools are routinely used to bypass the learning process, this system differs by promoting the positive use of AI to guide students towards true understanding. It does so by recommending targeted tutorials and quizzes that engage deeper learning rather than providing answers directly, thus enhancing the learning process itself. In addition, through the coupling of sequential knowledge tracing (LPKT) and graph-based neural networks (GNN), the model is made more aware of the temporal patterns of students' interactions with concepts and of relationships between concepts, enabling wiser, context-sensitive recommendations—e.g., refreshing related concepts when a student struggles with a particular topic—and generalizing more solidly across different learning paths. Since the model is trained on public datasets like EdNet: KT1 and uses commonly applied performance metrics, it has a very high probability of being implemented in mainstream educational systems like Khan Academy, Coursera, or Moodle,

where it can be used as an effective real-time learning system that facilitates personalized learning at a global level.

9 Discussion

The result of this study will provide concrete evidence for the application value of hybrid knowledge tracing systems in education AI. Combining high LPKT's learning and forgetting modeling ability with time and GNN's inter-concept relation defining ability, our model is able to gain a more complete understanding of student activities. These improvements can lead to more timely and relevant interventions, enhancing engagement and learning. Yet, the model's computational requirements and limited interpretability present practical challenges, especially for deployment in resource-constrained environments. Moreover, although performance metrics fortify predictive capability, they do not entirely encapsulate long-term learning improvements, which continue to be hard to measure. Subsequent versions need to venture into richer, multimodal inputs and more interpretable forms of AI to genuinely make personalized learning completely scalable and transparent.

10 Limitations

The model discussed here, although efficient and novel, suffers from several limitations and validity threats to which attention needs to be given. First, the combination of Learning Process Knowledge Tracing (LPKT) and Graph Neural Networks (GNN) has unparalleled computational complexity. The double architecture of the model consumes gigantic amounts of processing power and memory, especially when applied to large-scale datasets like EdNet containing over 100 million interactions. This increased resource demand manifests in increased training time and causes issues for deployment onto low-end hardware or mobile platforms, reducing availability in some learning contexts. Additionally, while the system recommends tutorials based on weak areas identified, it cannot generate dynamic, student-centered content. This limitation reduces the degree of personalization since students who are learning differently may require differing explanations even under the same subject area. Furthermore, the complexities of GNNs and LPKTs translate into lower levels of interpretability. This non-transparency is likely to prove a limitation within learning settings in which clarity will be crucial in teacher monitoring as well as students' confidence levels. Overfitting issues and generalization issues are big concerns as well; the model might overfit sparse data on students, and its performance would decline when being used on students whose behavior won't follow what it learned when trained. Dependence of the model on the well-labeled, structured data makes these issues worse. Careful concept labeling and frequent timestamping are required for optimum performance, but such highly detailed datasets would not always be available, particularly in less well-resourced educational systems.

Beyond such technical limitations, various threats to model validity must also be considered. Because the model is learned and tested only on the EdNet: KT1 dataset—extracted from one specific education platform—it might have an inherent bias limiting extrapolation to other subjects, curricula, or population contexts. Moreover, the mention of the generation of synthetic data for the sake of simulating learning patterns evokes the possibility of modelling artefacts that do not capture student behavior in the richness of the natural habitat. The measures of performance employed—accuracy, AUC, and RMSE—inform us about predictive accuracy but not about learning gain per se. may guess correctly and receive artificially high marks, and metrics like RMSE, as useful as they are, cannot distinguish systematic biases from random errors in prediction. Further, the model employs an 80/20 temporal train-test split and makes the assumption that student behavior is stationary over time. This assumption can be falsified by abrupt changes in a student's environment, i.e., new pedagogical methods or residential conditions, reducing the precision of the prediction model. Finally, the reproducibility threat has a friend with it if preprocessing operations and hyperparameter tuning are poorly documented or non-automated. This undermines the appropriateness of the model over time, especially scaling up to new data or learning environments, which ultimately limits its scope and scalability.

11 Threats to Validity

Several threats to the validity of the Smart Tutor system must be considered to fully understand the limitations of its current implementation and experimental design. One primary concern is dataset-specific bias, as the model is trained and tested solely on the EdNet: KT1 dataset, which, although extensive, originates from a single educational platform—Santa Learning. This concentrated source may lead the model to learn patterns that are only relevant to specific subjects, ages, curricula, or learning contexts and thus less precise when applied to predict larger or more diverse populations. Furthermore, the application of synthetic data to mimic learning patterns, as presented in the paper, has other risks. Artificial data sets, while useful for testing or augmentation, can generate artificial trends or simplifications that fail to represent variability and complexity in real student behavior, leading to over-optimistic or misleading measures of model performance. The second primary danger lies in the selection of evaluation metrics—namely, accuracy, AUC, and RMSE. While these measures quantify predictive performance, they are not necessarily reflective of genuine learning gains or knowledge retention. For instance, a correct guess can be made with no actual knowledge, exaggerating accuracy and AUC. Similarly, prediction error is measured by RMSE but cannot differentiate systematic bias from random error and tells us little about model reliability. The model's temporal train-test split assumption is another weakness. The chosen 80/20 ratio, as dictated by chronological time, assumes the behavior of students is moderately stationary across time. Practically, however, learning trajectories can be influenced by numerous external factors such as a change in pedagogy, life events, or motivational levels—variables that would undermine the predictive validity of the model. Finally, reproducibility threats also directly threaten the scalability and stability of the system. Without strict documentation and automation of preprocessing steps, hyperparameter tuning, and model parameters, it is hard to reproduce results or deploy the system to new datasets. This not only undermines the validity of the current findings but also hinders the long-term sustainability and adaptability of the Smart Tutor system in real-world educational settings.

12 Fututre Work

Future development of the Smart Tutor system offers ample possibilities for advancement, beginning with dynamic tutorial generation, where instead of merely recommending pre-existing material, future versions of the system may utilize the power of large language models (LLMs) or multimodal AI to generate on the fly personalized explanations or visualizations based on a student's current knowledge state and preferred learning style—textual, visual, gamified, or otherwise. To further the contextual awareness of the system, future models need to engage with multimodal student data beyond the present sole reliance on quizzes and timestamps. This can include text-based input in the form of essay responses or search questions, audio/video-based interaction in the form of spoken response or video-recorded tutoring, and affective input in the form of facial emotion detection to capture emotional states like frustration or confidence. Furthermore, inclusion of continuous capacity to learn would ensure the model change in real time as the students grow, and for this purpose, it would require online learning algorithms, real-time feedback cycles, and confidence-driven recommendation updates. Explainable AI would be a highest-priority feature in future releases, wherein introducing visualizations and natural language descriptions of each recommendation would push transparency to a completely different level, allowing students to understand the reasons behind the system and give teachers the option of interpreting or stepping in accordingly. Cross-disciplinary application is also an essential area of growth, with the model being extended to domains like language learning, where knowledge is less discrete and more fluid, or skill-based areas like programming or music, which require different evaluation frameworks. Improving the underlying graph learning techniques is another vital direction; instead of relying on static structures, future work could introduce dynamic graph updates based on evolving student performance trends and explore meta-learning strategies to let the graph models self-adapt to varying educational domains or learner types. Finally, ensuring the system's scalability and accessibility will be key, which means optimizing the model to function efficiently on edge devices or in environments with limited bandwidth and making it usable in underrepresented or resource-constrained regions—ultimately broadening the reach of personalized, intelligent education.

References

[1] Yang, Yang, et al. "GIKT: a graph-based interaction model for knowledge tracing." Machine learning and knowledge discovery in databases: European conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, proceedings, part I. Springer International Publishing, 2021.

[2] Xiao, Jun, et al. "A personalized recommendation system with combinational algorithm for online learning." Journal of ambient intelligence and humanized computing 9 (2018): 667-677.

[3] Shen, Shuanghong, et al. "Learning process-consistent knowledge tracing." Proceedings of the 27th ACM SIGKDD conference on knowledge discovery and data mining. 2021.

[4] Yang, Shangshang, et al. "Evolutionary neural architecture search for transformer in knowledge tracing." Advances in Neural Information Processing Systems 36 (2023): 19520-19539.

[5] Long, Ting, et al. "Tracing knowledge state with individual cognition and acquisition estimation." Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021.