LLM-driven Effective Knowledge Tracing by Integrating Dual-channel Difficulty

Jiahui Cen[1], Jianghao Lin[1,2*], Dong Zhou[1], Weixuan Zhong[1], Jin Chen[3], Aimin Yang[4], Yongmei Zhou[1*]

[1]School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, 510000, Guangdong, China.
[2]Laboratory for Language Engineering and Computing, Guangdong University of Foreign Studies, Guangzhou, 510000, Guangdong, China.
[3]School of Foreign Languages, South China University of Technology, 381 Wushan Road, Guangzhou, 510641, Guangdong, China.
[4]School of Computer Science and Intelligence Education, Lingnan Normal University, Zhanjiang, 524048, Guangdong, China.

Abstract

Knowledge Tracing (KT) is a fundamental technology in intelligent tutoring systems used to simulate changes in students' knowledge state during learning, track personalized knowledge mastery, and predict performance. However, current KT models face three major challenges: (1) When encountering new questions, models face cold-start problems due to sparse interaction records, making precise modeling difficult; (2) Traditional models only use historical interaction records for student personalization modeling, unable to accurately track individual mastery levels, resulting in unclear personalized modeling; (3) The decision-making process is opaque to educators, making it challenging for them to understand model judgments. To address these challenges, we propose a novel Dual-channel Difficulty-aware Knowledge Tracing (DDKT) framework that utilizes Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) for subjective difficulty assessment, while integrating difficulty bias-aware algorithms and student mastery algorithms for precise difficulty measurement. Our framework introduces three key innovations: (1) Difficulty Balance Perception Sequence (DBPS) - students' subjective perceptions combined with objective difficulty, measuring gaps between LLM-assessed difficulty, mathematical-statistical difficulty, and students' subjective perceived difficulty through attention mechanisms; (2) Difficulty Mastery Ratio (DMR) - precise modeling of student mastery levels through different difficulty zones; (3) Knowledge State Update Mechanism

- implementing personalized knowledge acquisition through gated networks and updating student knowledge state. The DDKT framework dynamically adjusts question difficulty based on students' current knowledge state and learning progress, ensuring optimal challenge levels for everyone. Experimental results on two real datasets show our method consistently outperforms nine baseline models, improving AUC metrics by 2% to 8% while effectively addressing cold-start problems and enhancing model interpretability.

Keywords: Knowledge Tracing, Large Language Models, Cold-start Problem, Difficulty-aware Learning, Personalized Student Modeling

1 Introduction

Knowledge Tracing (KT) has become an indispensable component in intelligent tutoring systems, utilizing students' historical response data to assess their knowledge mastery and automatically predict their future performance and knowledge state[1]. In the educational domain, knowledge tracing can optimize students' learning paths and provide guidance for teachers, enabling targeted improvements in teaching strategies and personalized instruction. Through dynamic tracking of students' knowledge state, this technology effectively reduces teachers' workload while enhancing teaching quality. Fig 1 demonstrates a simple overview of Knowledge Tracing. By analyzing two students' performance across five mathematical problems, we observe distinct learning patterns. Student 1 shows an alternating pattern of correct and incorrect responses (correct-incorrect-correct-incorrect) in the first four questions, achieving a 50% accuracy rate. In comparison, student 2 demonstrates a more stable learning trajectory with a 75% accuracy rate in the first four questions, showing consistent improvement after an initial incorrect response in multiplication. Both LLM-based and Statistical-based difficulty assessments indicate that the final subtraction problem (difficulty: 59/58) is slightly less challenging than the previous subtraction problem (65/60). Considering the learning patterns, performance trajectories, and difficulty assessments, we predict that student 1 is likely to respond incorrectly to the final question, while student 2 is likely to provide a correct response.

With the rapid development of online education, vast amounts of learning behavior data have been recorded, encompassing not only the correctness of students' answers but also multi-dimensional information such as response time and learning resource utilization patterns. This abundance of data has established a solid foundation for advancing knowledge tracing technology. However, effectively utilizing this data to accurately assess students' knowledge state while balancing objective difficulty with students' subjective learning conditions remains a challenging issue. In recent years, the field of knowledge tracing has witnessed the emergence of numerous advanced deep learning models, broadly categorized into Deep Knowledge Tracing models (e.g., DKT[2], which revolutionized the field in 2015), Attention-based models (e.g., SAKT[3], AKT[4], which capture long-term dependencies), Memory-based

Fig. 1 A Simple Overview of Knowledge Tracing Based on LLM Difficulty Assessment and Statistical Difficulty Assessment.

models (e.g., DKVMN, which maintains explicit memory representations), Graph-based models (e.g., GKT[5], HGKT[6], which model knowledge concept relations), and Probabilistic Analysis models. While these models have achieved significant performance improvements, they have also introduced new challenges, particularly in terms of model interpretability. Due to the black-box nature of deep learning, the decision-making process lacks transparency and interpretability. Although this end-to-end learning strategy has improved prediction accuracy, the models lack theoretical guidance in calculating students' knowledge state. Without clear explanatory mechanisms, any erroneous predictions may lead to students' distrust in the system, thereby limiting these models' applications in practical educational scenarios.

With the gradual increase in computational power in recent years, LLMs have emerged prominently across various fields. In knowledge tracing, these models can comprehend deep semantic information within questions and extract rich side information as new features for traditional models. In traditional knowledge tracing research, objective difficulty is primarily based on statistical data and question characteristics, which influences learning resource allocation and learning path planning. Subjective difficulty, on the other hand, needs to consider factors such as students' personal characteristics, learning styles, and knowledge mastery levels, but few studies have effectively combined these two difficulty dimensions. Addressing this research gap, our proposed DDKT framework innovatively integrates objective difficulty assessment with subjective difficulty perception based on students' personalized characteristics, achieving more precise difficulty evaluation and personalized learning experiences through LLMs and retrieval-augmented generation technology.

We employ three new techniques to optimize the model's prediction steps. First, we use Large Language Models (LLMs) to assess question difficulty and combine objective statistical difficulty with students' subjective difficulty perception to construct student difficulty perception bias. Second, we adopt more refined student difficulty perception modeling, enabling the model to better understand the relationships between questions of similar difficulty. Third, we construct a prediction model based on gating mechanisms and Transformer architecture that can process and understand long-term dependencies in sequences. More specifically, learning state improve when objective question difficulty aligns more closely with students' perceived difficulty. This is because when there is a significant disparity between objective difficulty and student-perceived difficulty, students may engage in guessing or making mistakes (guess and slip).

The main contributions of this paper can be summarized as:

- We innovatively combine LLM difficulty assessment with statistical difficulty assessment, integrating students' subjective difficulty perception as a new feature, effectively mitigating the cold-start problem in knowledge tracing. Through LLM's semantic understanding capabilities, educators can more intuitively understand the model's difficulty assessment process, significantly enhancing model interpretability and enabling more targeted teaching decisions.
- We propose the Difficulty Mastery Rate (DMR) algorithm, which constructs more accurate personalized knowledge state profiles through fine-grained modeling of student performance patterns across different difficulty intervals. This method significantly improves the model's evaluation metrics, making prediction results more aligned with students' actual learning state.
- Our proposed prediction model presents an architecturally sophisticated design, leveraging an advanced gated Transformer architecture to effectively capture long-term dependencies in sequences. The strategic integration of attention mechanisms with gated networks enhances the model's expressiveness, resulting in superior prediction performance while maintaining optimal resource utilization.
- Extensive experiments on two public datasets demonstrate that our proposed DDKT framework achieves state-of-the-art (SOTA) performance compared to existing deep learning-based knowledge tracing models. The experimental results validate the model's significant advantages in terms of accuracy, efficiency, and interpretability.

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 presents the overall architecture and specific steps of our model, Section 4 discusses the experiments and their results, and Section 5 concludes the paper.

2 Related Work

2.1 Deep Learning-Based Knowledge Tracing

Knowledge tracing is essentially a time series classification task that predicts student performance based on their historical learning records. Early knowledge tracing research primarily focused on combining Bayesian Knowledge Tracing (BKT)[7] or

using Item Response Theory (IRT)[8]. With the flourishing development of deep learning, numerous innovative models have emerged in this field. Piech et al. proposed Deep Knowledge Tracing (DKT), as the first model to introduce deep learning into knowledge tracing, employing a Recurrent Neural Network (RNN) architecture to capture temporal relationships in student interaction data[2]. This pioneering work not only significantly improved prediction performance but also laid a crucial foundation for subsequent research. Compared to traditional methods, DKT can automatically extract features from large-scale learning behavior data and effectively handle complex relationships among multiple knowledge concepts. Chen et al. proposed QIKT based on the DKT model, introducing question-centric knowledge acquisition, knowledge state, and problem-solving modules, breaking the traditional homogeneous question assumption[9]. The model innovatively combines IRT-based interpretable prediction layers, providing better interpretability while maintaining high prediction performance.

Following DKT, memory-based models gradually emerged, with Dynamic Key-Value Memory Networks (DKVMN) proposed by Zhang et al. being the most representative. This model enhances the representation of student historical records through a key-value pair mechanism, more accurately capturing students' latent knowledge state[10]. DKVMN utilizes explicit memory representations to precisely track students' mastery of different knowledge concepts, and its unique memory mechanism optimizes the modeling of long-term dependencies.

However, when facing sparse learning data scenarios (such as students interacting with only a few knowledge concepts), traditional models struggle to effectively capture implicit relationships within the data. To address this issue, attention-based models were developed. Liu et al. proposed AT-DKT based on the DKT model, introducing Question Tag (QT) prediction and Individual prior Knowledge (IK) prediction as auxiliary learning tasks[11]. The model uses Transformer encoders and masked attention mechanisms for QT tasks and student ability networks for IK tasks. Pandey et al. proposed Self-Attentive Knowledge Tracing (SAKT), which through its multi-head attention mechanism and feed-forward network, can identify knowledge concepts (KCs) related to the target KC from historical activities and predict student mastery based on limited historical data[3]. Compared to RNN-based approaches, SAKT demonstrates significant advantages in handling data sparsity issues. Huang et al. proposed sparseKT, a simple yet effective framework aimed at improving the robustness of attention-based knowledge tracing models[12]. The model's core innovation lies in introducing k-sparse attention mechanisms, making predictions by explicitly selecting the most relevant historical interactions.

Taken together, the development of knowledge tracing models has evolved from traditional probabilistic models to deep learning approaches. However, how to better balance model performance, interpretability, and computational efficiency, as well as how to effectively integrate students' personalized characteristics, remain important directions for further exploration in this field. In traditional knowledge tracing models, predictions are mostly based on students' learning history records, such as problem IDs, knowledge concept IDs, and whether students are correct, without incorporating difficulty features into the training process. Therefore, it is impossible to combine

students' knowledge state and potential difficulty features of questions to build a more refined prediction model. Similarly, the lack of difficulty features also weakens the model's interpretability, specifically manifested in: 1) educators cannot accurately evaluate students' knowledge mastery level based on their performance on questions of different difficulties, leading to a lack of targeted teaching strategy adjustments; 2) the model cannot distinguish students' performance differences on questions with the same knowledge points but different difficulties, affecting the precise analysis of learning trajectories; 3) when model predictions deviate from actual performance, it is difficult to explain the reasons for such deviations from the perspective of difficulty, reducing the credibility and practicality of prediction results.

2.2 Difficulty in Knowledge Tracing

Question difficulty, as a crucial evaluation metric in the educational process, can serve as a special predictive feature in knowledge tracing[13]. In recent years, research incorporating additional information such as difficulty has gradually increased, though most approaches employ single difficulty information[14].

DIMKT is a difficulty-aware knowledge tracing model derived from response correctness rates, which improves knowledge tracing performance by establishing relationships between students' knowledge state and question difficulty levels to measure the difficulty effect[15]. The model's innovation lies in enhancing question representation by simultaneously considering both question-specific difficulty and knowledge concept difficulty, and designing three stages to capture the difficulty effect: first calculating students' subjective difficulty perception before practice, then estimating students' personalized knowledge acquisition when answering questions of different difficulty levels, and finally updating students' knowledge state according to question difficulty.

Similarly focusing on difficulty, Liu et al. proposed the QDCKT model, which combines first-attempt correctness rate-based difficulty with graph attention mechanism, innovatively replacing traditional question IDs with question difficulty levels[16]. The model employs an LSTM sublayer to generate representations of historical learning sequences and uses a feed-forward neural network as the prediction layer. QDCKT introduces two key techniques: first using the Hann function to combine embeddings of nearby difficulty levels, and second introducing difficulty consistency constraints to ensure prediction results align with question difficulty levels.

Zhang et al. proposed GDPKT based on the Graph-based Knowledge Tracing (GKT) model[17]. It enhances knowledge tracing capabilities through heterogeneous graph neural networks and personalized difficulty modeling. The model introduces difficulty nodes into the heterogeneous graph, uses Meta-path to construct node representations, and combines difficulty perception and learning gain modules to model students' knowledge state personalized. The main innovation lies in modeling exercise difficulty as independent nodes in the graph while considering students' personalized perception and learning gains from exercises of different difficulties. Qiu et al. proposed MGEKT is a knowledge tracing model based on multi-graph embedding[18]. It adopts a dual-channel architecture, with one channel using node2vecWalk and Meta-path to enhance question representation, and the other channel using AGCN to process

directed graphs of learning interactions. The model innovatively considers question difficulty from three dimensions (correctness rate, attempt count, and response time) and introduces reverse knowledge distillation to integrate information from both channels. Experiments demonstrate that this model outperforms existing methods across four datasets, significantly improving knowledge tracing performance.

However, relying solely on statistically derived difficulty as a question difficulty indicator often lacks objectivity and comprehensiveness. This manifests in the data when limited student responses to certain questions result in extremely high or low difficulty values. Moreover, statistical difficulty fails to effectively capture intrinsic question characteristics such as concept complexity and solution steps. This single-dimensional difficulty measurement approach limits the precise modeling of students' learning state in knowledge tracing models. Due to these limitations, existing models often encounter cold-start problems when dealing with questions having few interaction records, making it challenging for difficulty features to serve as objective indicators for learning the intrinsic connections between difficulty and questions. This not only affects model prediction accuracy but also limits its practical application in educational scenarios. Furthermore, the multi-dimensional nature of difficulty assessment has not received sufficient attention in current research. Question difficulty depends not only on statistical data but should also consider solution complexity, knowledge structure complexity, and other dimensions, which opens possibilities for introducing new technologies like large language models to enhance the objectivity and multi-dimensionality of difficulty assessment.

2.3 Large Language Model Application in Knowledge Tracing

In recent years, the field of Natural Language Processing has witnessed significant technological advancement with the emergence of increasingly sophisticated tools. Large Language Models (LLMs), as representatives of new-generation pre-trained models, have demonstrated exceptional performance across various domains. Models like ChatGPT and Claude, utilizing techniques such as Chain-of-Thought (COT), have substantially enhanced their capabilities in understanding and generating deep semantic information in natural language. However, as general-purpose language understanding and generation tools, LLMs show limitations in handling strong sequential problems in knowledge tracing, thus primarily serving as auxiliary tools in knowledge tracing tasks[19]. Yu et al. proposed ECKT, a knowledge tracing model based on large language models and CodeBERT[20]. It generates question descriptions and knowledge concepts through chain-of-thought reasoning and few-shot learning, uses BERT for embedding, and combines AST and attention mechanisms for code representation. ECKT innovates by using LLM to generate question descriptions and knowledge concepts, introducing difficulty embeddings to enhance representation, and employing stacked GRU to improve sequence learning capabilities. Guo et al. proposed EAKT (Enhanced Attribute-aware Knowledge Tracing), a LLM-based cold-start solution for knowledge tracing that leverages LLMs to extract additional information such as required problem-solving capabilities[21]. EAKT introduces three key innovations: first, it designs an attribute estimation module that utilizes models like GPT-4 with grouping strategies and chain-of-thought prompting to analyze questions and estimate

Symbol	Meaning
q_i	Question i
c_i	Concept of question i
r_i	Student's response of question i
$var_t^{indicator}$	Indicator of variable in timestamp t, e.g. Embedding
var_i^t	Variable of question i in timestamp t
ks_t	Student's knowledge state in timestamp t
d_t^{type}	Difficulty of question sequence in timestamp t, e.g. LLM-based Difficulty
dmr_t	Difficulty Mastery Ratio of question sequence in timestamp t
$dpbs_t$	Difficulty Perception Bias Sequence of question sequence in timestamp t
$ddai_t$	Dynamic Difficulty Adaptability Index of question sequence in timestamp t

Table 1 Mathematical symbol and its meanings.

multi-dimensional attributes including difficulty, ability requirements, and expected response time; second, it develops a question embedding module that employs graph attention networks to dynamically adjust attribute values for alignment with the target population's cognitive characteristics; finally, it enhances model interpretability through multi-dimensional attribute representations.

Lee et al. introduced the DCL4KT+LLM (Difficulty-focused Contrastive Learning for Knowledge Tracing with LLM) model, which integrates difficulty-focused contrastive learning with LLM-based difficulty prediction[16]. The model comprises three core components: first, it constructs positive and negative embedding layers, where positive embeddings contain question, concept, question difficulty, concept difficulty, and student response information, while negative embeddings contain corresponding "hard negative" information; second, it employs four encoder modules to calculate binary cross-entropy loss and contrastive learning loss separately; third, it introduces a difficulty-focused contrastive learning framework that optimizes model representations through concept and question similarity computations. Yang et al. proposed DPKT, a programming knowledge tracing model based on large language models and CodeBERT[22][23]. It uses large language models to evaluate both the text comprehension difficulty and knowledge concept difficulty of programming questions, extracts semantic features through CodeBERT, and combines graph attention networks with update gate mechanisms to dynamically adjust students' knowledge state. Its innovation lies in being the first to use large language models to assess programming question difficulty, dividing difficulty into text comprehension and knowledge concept dimensions for more precise difficulty assessment.

While current research has shown promising results in utilizing LLMs to extract supplementary information for knowledge tracing, many existing methods simply incorporate LLM-extracted additional information directly into the input, overlooking individual students' difficulty perception. Therefore, how to effectively integrate objective difficulty assessment with personal difficulty perception to enhance knowledge tracing model performance remains an important research direction. Incorporating personal difficulty perception into objective question difficulty could enable more fine-grained student modeling and improve model prediction accuracy.

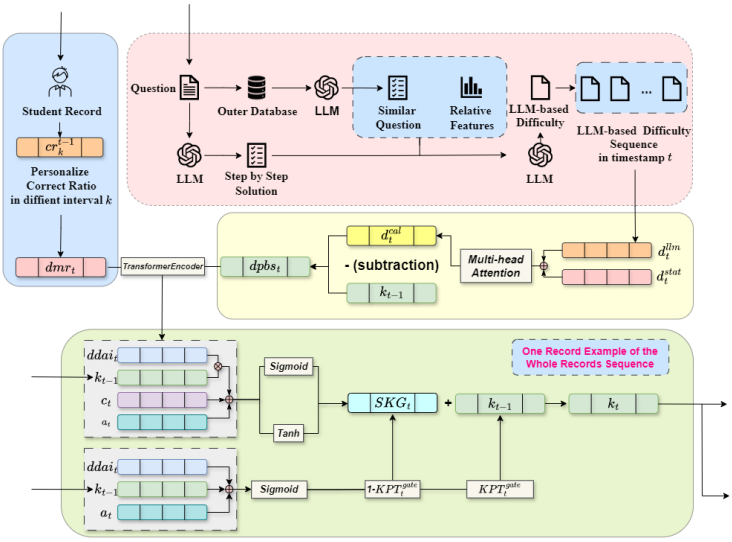

Fig. 2 Architecture of the DDKT model, consisting of two parts: (upper) three feature extraction modules including personalized correct ratio for DMR, LLM-based difficulty assessment, and knowledge state update with DPBS; (lower) dual-channel difficulty-aware knowledge tracing model combining engagement features and knowledge state features to generate final knowledge predictions.

3 Proposed Method

3.1 Problem Definition

Knowledge Tracing (KT) is a task that predicts students' future performance based on their historical interaction records in Learning Management Systems (LMS) or Intelligent Tutoring Systems (ITS). By analyzing students' learning trajectories, knowledge tracing models can assess students' knowledge state in real-time, thereby supporting personalized learning. The formal definition is as follows: Let $S = \{s_1, s_2, ..., s_n\}$ be the set of students, $Q = \{q_1, q_2, ..., q_m\}$ be the set of questions, $C = \{c_1, c_2, ..., c_k\}$ be the set of concepts, and $R = \{0, 1\}$ be the set of responses, where 0 indicates an incorrect answer and 1 indicates a correct answer. For student s_i, their interaction sequence can be represented as $x_i = \{x_{i1}, x_{i2}, ..., x_{it}\}$, where each interaction x_{it} is a triple (q_{it}, c_{it}, r_{it}) representing the question attempted, the concept involved, and the response result at timestamp t, respectively. The objective of knowledge tracing is to predict the probability $P(r_{i(t+1)} = 1|x_{i1}, x_{i2}, ..., x_{it})$ that a student will correctly answer a new question at timestamp $t+1$, based on their historical interaction sequence $\{x_{i1}, x_{i2}, ..., x_{it}\}$. This is a typical sequential prediction problem that requires the model to effectively capture the dynamic changes in students' knowledge state.

3.2 Utilizing LLMs to determine the question's difficulty

To extract the implicit difficulty information from questions, we designed a specialized prompt that enables LLMs to perform similar question retrieval and difficulty

assessment. This process employs the Chain of Thought approach[24], conducting step-by-step analysis across different problem-solving stages to ultimately determine the question's difficulty level.

Phase 1 *(Step-by-Step Solution Generation by LLM)*: In this stage, we employ a Step by Step Solution Prompt. This input prompt guides the LLMs through the problem-solving process, allowing us to directly observe the complexity of the solution process. The input contents include:

- **Problem Context**: The question's original or restructured context
- **Problem Answer**: The question's ground-true answer
- **Brief Explanation**: The brief explanation of how to solve the question(if existed in dataset, e.g. XES3G5M)
- **Problem Type**: The problem type of the question(Fill in the Blank or multiple choice questions)
- **Options (if applicable for multiple choice questions)**: The options if the problem type is multiple choice, otherwise will be empty.

These input elements are incorporated into a well-structured step-by-step solution prompt. Through this approach, the LLMs generates clear solution steps. This structured problem-solving process helps assess not only the cognitive complexity of the problem but also reveals the depth of knowledge and reasoning difficulty required. This method is particularly effective in identifying questions that appear simple but involve complex solution processes, or vice versa, thus providing reliable baseline data for subsequent difficulty assessment.

Phase 2 *(Extract Difficulty by LLM)*: After obtaining the step-by-step solution from the previous phase, we employ a difficulty assessment prompt in Phase 2. This prompt guides the model to evaluate the question's difficulty based on the text, knowledge concepts, and other information contained in the question. The input contents include:

- **Problem Context, Brief Explanation, and Problem Type**: Same as *Phase 1* inputs
- **Knowledge Concepts (KCs)**: The knowledge concepts contained in the question, including hierarchical refinement of knowledge concepts
- **Step-by-Step Solution**: The output from *Phase 1*

The output will be the difficulty assessment value evaluated by LLM based on the input contents, denoted as d_i^{llm}.

After these two phases, the large language model generates an LLM-based difficulty assessment, which serves as a crucial component in the objective difficulty dimension. By employing the Chain of Thought technique, the model can analyze various aspects of the question through explicit reasoning processes, including concept complexity, number of solution steps, and knowledge point correlations, resulting in more accurate and interpretable difficulty assessments. This fine-grained analysis-based evaluation method not only improves the accuracy of difficulty judgment but also provides educational practitioners with specific difficulty attribution evidence. Compared to traditional statistics-based assessment methods, this approach better

captures the latent characteristics of question difficulty, showing particular advantages when dealing with new questions or scenarios lacking extensive historical data.

3.3 Statistical and LLM-based Difficulty Calibration

In addition to LLM-assessed difficulty, statistical difficulty serves as another crucial indicator in our framework. Statistical difficulty is denoted as d_i^{stat}, represents an empirical measurement derived from historical student performance data, specifically using the correct response rate of all students for a particular question as the statistical difficulty indicator. The statistical difficulty for question i can be expressed by the following formula:

$$d_i^{stat} = \frac{\sum_{j=1}^{N} r_{ij}}{N} \tag{1}$$

where d_i^{stat} represents the statistical difficulty of question i, r_{ij} represents student j's response to question i (1 for correct, 0 for incorrect), N represents the total number of students who answer question i and $\frac{\sum_{j=1}^{N} r_{ij}}{N}$ represents the correct response rate.

This formula directly uses the correct response rate as the statistical difficulty indicator. A higher correct rate indicates lower difficulty, while a lower correct rate indicates higher difficulty. The difficulty value ranges from 0 to 1, where 1 indicates the easiest (all students answered correctly) and 0 indicates the most difficult (all students answered incorrectly).

To leverage the complementary strengths of both approaches, we integrate statistical difficulty with LLM-assessed difficulty using a multi-head attention mechanism. This integration produces a calibrated difficulty score (d_i^{cal}) that synthesizes the data-driven statistical insights with the semantic understanding capabilities of large language models. The multi-head attention mechanism enables our model to dynamically weight different aspects of both difficulty measurements, resulting in a more comprehensive and robust difficulty assessment that captures both empirical patterns and theoretical complexity. Thus, the formula of d_i^{cal} is defined as

$$d_i^{cal} = MultiHeadAttention(d_i^{stat}, d_i^{llm}) \tag{2}$$

After calculating d_i^{cal}, it is mapped to question i. During the training process, each time series has d_t^{cal}, which represents the calibrated difficulty sequence corresponding to questions within that time series.

3.4 Difficulty Perception Bias Sequence

During the learning process, students' knowledge state often reflect their subjective perception of difficulty. When a student has a low mastery level of a certain concept, their knowledge state for that concept tends to be low. Based on this observation, we propose the concept of Difficulty Perception Bias Sequence (DPBS). The DPBS in timestamp t is calculated as

$$dpbs_t = d_t^{cal} - ks_{t-1} \tag{3}$$

where d_t^{cal} represents the objective difficulty of the question assessed by the large language model at timestamp t, and ks_{t-1} represents the student's knowledge state at timestamp $t-1$. This difference reflects the degree of bias between a student's subjective difficulty perception and the objective difficulty of the question.

When DPBS is close to zero ($dpbs_t \approx 0$), it suggests an optimal match between question difficulty and student knowledge level, representing the ideal learning state. When DPBS is significantly positive ($dpbs_t \gg 0$), it indicates that the objective difficulty of the question far exceeds the student's current knowledge level, typically occurring when students with weaker foundations encounter challenging questions. When DPBS is significantly negative ($dpbs_t \ll 0$), it shows that the objective difficulty is much lower than the student's knowledge level, usually occurring when high-performing students solve basic questions.

3.5 Difficulty Mastery Ratio

In the process of personalized student modeling, the mastery ratio of students across different difficulty levels reveals distinctive patterns among individuals, which is crucial for personalized modeling. To effectively capture these individual differences, we propose the Difficulty Mastery Ratio (DMR), a novel metric that quantifies students' mastery patterns across varying difficulty levels, thereby enabling more precise personalized modeling of student learning characteristics.

We first partition the difficulty range [0,100] into five equal intervals, each representing a distinct difficulty level. Based on the statistical difficulty d_i^{stat}, we assign each question i to its corresponding difficulty interval and calculate the correct response rate within each interval. For student interaction data $x_{i(t-1)}$, the correct response rate cr_k^{t-1} of each difficulty interval is appended to $x_{i(t-1)}$. Considering the natural occurrence of knowledge decay in the learning process, we introduce a forgetting factor σ to simulate the gradual forgetting process in student learning. The Difficulty Mastery Ratio (DMR) at timestamp t can be expressed by the following formula

$$cr_k^{t-1} = \frac{\sum_{i \in B_k} r_i^{t-1}}{\left|B_k^{t-1}\right|} \tag{4}$$

$$dmr_k^t = \sigma \cdot cr_k^{t-1} \tag{5}$$

where B_k represents the kth difficulty interval (k = 1,2,3,4,5), r_i^t represents the student's response to question i at timestamp t (1 for correct, 0 for incorrect), $\left|B_k^{t-1}\right|$ represents the number of questions in interval k at timestamp t and $\frac{\sum_{i \in B_k} r_i^{t-1}}{\left|B_k^{t-1}\right|}$ represents the current correct rate in the difficulty interval.

3.6 Dynamic Difficulty Adaptability Index

To comprehensively evaluate students' learning characteristics, we propose integrating DPBS and DMR to construct the Dynamic Difficulty Adaptability Index (DDAI). DPBS combines calibrated difficulty with students' subjective difficulty assessments, providing a comprehensive difficulty measure that encompasses both objective evaluation and subjective perception. Meanwhile, DMR captures students' mastery levels

across different difficulty tiers, offering crucial insights for personalized modeling. This integration leverages the complementary advantages of both metrics: DPBS reflects the comprehensiveness of difficulty assessment, while DMR embodies the personalized characteristics of the learning process. By combining these two dimensions, DDAI can more accurately describe students' adaptability and learning characteristics when facing questions of varying difficulty levels. We employ a TransformerEncoder that utilizes its multi-head attention mechanism to capture dynamic features and long-term dependencies in both DPBS and DMR sequence data. The multi-head attention mechanism enables the model to fully comprehend the complex interaction patterns between students' difficulty perception bias (DPBS) and mastery levels (DMR) across different timestamp. The TransformerEncoder takes the sequence data of DPBS and DMR as input, preserves temporal information through positional encoding, and leverages self-attention mechanisms to learn key patterns within the sequences, effectively integrating the dynamic changes in students' difficulty perception and mastery levels. Specifically, the DMR value for question i is formula as

$$dmr_i^t = dmr_k^t \tag{6}$$

where the statistical difficulty of question i is in $(k, k+20]$. Since the students' interaction records are a time series, we will calculate dmr_i^t for each question i at timestamp t, resulting in the DMR sequence dmr_t for timestamp t. After that, the DDAI value for a student at timestamp t is calculated as

$$ddai_t = TransformerEncoder(dpbs_t, dmr_t) \tag{7}$$

3.7 Dual-channel Difficulty Knowledge Tracing Model

Traditional knowledge tracing models typically rely solely on statistical difficulty (such as correct rates) when evaluating question difficulty, which presents several significant limitations: First, statistical difficulty is susceptible to factors such as sample size and data distribution imbalance, leading to insufficient objectivity in difficulty assessment; Second, this approach ignores individual students' subjective perception differences regarding question difficulty, failing to accurately reflect different students' learning characteristics; Third, static difficulty assessment cannot capture students' ability improvement and adaptability changes during the learning process. To overcome these limitations, we propose a more comprehensive solution: introducing Large Language Models (LLM) for question difficulty analysis, combining statistical difficulty with student subjective assessments to construct a multi-dimensional difficulty evaluation system. This approach not only enhances the objectivity and reliability of difficulty assessment but also enables personalized learning models for each student, more accurately predicting learning trajectories and performance.

Student Knowledge Gain. To simulate student learning behavior and integrate both objective question difficulty and students' subjective difficulty perception, we designed a novel knowledge tracing model DDKT. As mentioned above, after extracting $ddai_t$, we obtain its embedding vector $ddai_t^{emb}$. Subsequently, we perform element-wise multiplication between $ddai_t^{emb}$ at timestamp t and the student's

knowledge state ks_{t-1} at timestamp $t-1$ to obtain the input embedding. This element-wise multiplication design is based on the following consideration: by multiplying dynamic difficulty assessment values with knowledge state, the model can capture the interaction between current knowledge levels and question difficulty, thus more accurately simulating student performance variations when facing questions of different difficulties. This design enables the model to dynamically adjust the weights of question difficulty based on students' knowledge state, better reflecting personalized characteristics in the learning process. The input data $input_t$ is formula as

$$input_t = ddai_t^{emb} * ks_{t-1} \tag{8}$$

where $*$ represents the element-wise multiplication for two vectors.

After the input embedding $input_t$ is calculated, it is combined with the embedding of the student's response sequence r_t^{emb} and the embedding of knowledge concept of questions sequence c_t^{emb}. The input data is calculated as

$$z_t = input_t \oplus r_t^{emb} \oplus c_t^{emb} \tag{9}$$

where z_t is the new input data as it is calculated by $input_t$, r_t^{emb} and c_t^{emb}, $\oplus$ represented as the concatenate operator.

The use of z_t instead of solely relying on $input_t$ stems from the fact that the interaction between difficulty assessment and knowledge state alone may not fully reflect students' actual learning situations. Students' specific answer performance contains important real-time feedback information. On one hand, students' actual responses reflect their current knowledge mastery level; on the other hand, combining difficulty assessment information can more accurately interpret student performance, thus providing more reliable basis for subsequent knowledge state updates. The calculation of Student Knowledge Gain (SKG_t) is shown in Equation 10.

$$\begin{aligned} SKG_t^{val} &= tanh(z_t) \\ SKG_t^{gate} &= sigmoid(z_t) \\ SKG_t &= SKG_t^{val} * SKG_t^{gate} \end{aligned} \tag{10}$$

here SKG_t^{val} represented as the Student Knowledge Gain value, SKG_t^{gate} represented as the gate of Student Knowledge Gain for controlling the output information in SKG_t^{val}. SKG_t represented the output value of Student Knowledge Gain.

Student Knowledge Update. After obtaining the knowledge acquisition amount following the student's exercise completion, the model updates the student's knowledge state from timestamp $t-1$. Similarly, we employ a gating mechanism to calculate the student's knowledge state at timestamp t. Specifically, we calculate the Knowledge Pass Thought (KPT_t) at timestamp t, which controls the amount of knowledge state passage from timestamp $t-1$, and combines it with the knowledge acquisition amount at timestamp t to derive the specific knowledge state at timestamp t. To accomplish this task, we consider using the dynamic difficulty adaptability index $ddai_t^{emb}$ at timestamp t, its knowledge state at timestamp $t-1$, and the response at timestamp t

as inputs, with the output being the student's knowledge state at timestamp t. The calculation formula is shown in Equation 11.

$$\begin{aligned} KPT_t^{gate} &= sigmoid(ddai_t^{emb} \oplus ks_{t-1} \oplus r_t^{emb}) \\ ks_t &= KPT_t^{gate} * ks_{t-1} + (1 - KPT_t^{gate}) * SKG_t \end{aligned} \tag{11}$$

where ks_t is the output value, represented as the student knowledge state in timestamp t. KPT_t^{gate} is a gate mechanism to control the information pass thought ratio in ks_{t-1} and SKG_t.

3.8 Prediction Layer And Objective Function

After obtaining the student's knowledge state ks_t at timestamp t, we can predict the student's performance at timestamp $t+1$ based on ks_t. In our model, the input data consists of the student's knowledge state ks_t at timestamp t multiplied by the knowledge representation embedding $target_{t+1}^{emb}$ at timestamp $t+1$, with a sigmoid function outputting the probability of correct response. The knowledge representation embedding $target_{t+1}^{emb}$ includes the question q_t, concept c_t, statistical assessment difficulty d_t^{stat}, and large language model assessment difficulty d_t^{llm} for timestamp $t+1$. Specifically, y_{t+1} can be expressed as

$$\begin{aligned} d_{t+1}^{emb} &= Embedding(d_{t+1}^{llm}) \oplus Embedding(d_{t+1}^{llm}) \\ target_{t+1}^{emb} &= q_{t+1}^{emb} \oplus c_{t+1}^{emb} \oplus d_{t+1}^{emb} \\ y_{t+1} &= sigmoid(ks_t * target_{t+1}^{emb}) \end{aligned} \tag{12}$$

where $Embedding(x)$ represents the embedding value of element x, $target_{t+1}^{emb}$ is the question representation, and y_{t+1} is the output prediction probability value. We use Binary Cross Entropy as the loss function, the function can be formula as

$$\ell_{loss} = \frac{\sum_{n=1}^{N} -[y_n \cdot \log(x_n) + (1 - y_n) \cdot \log(1 - x_n)]}{N} \tag{13}$$

where x_n represents the groundtruth value, y_n represents the predicted value, and N represents the total number of samples.

4 Experiment

4.1 Dataset

We selected two real-world public datasets to train and evaluate our model: XES3G5M and Eedi. The detailed information of the two datasets is shown below:

- ***XES3G5M*** is a large-scale dataset that contains numerous questions and auxiliary information about related knowledge components (KCs)[25]. The XES3G5M dataset was collected from a real online mathematics learning platform and includes 7,652 questions, 865 KCs, and 5,549,635 learning interactions from 18,066 students.

- ***Eedi (NIPS34)*** dataset comes from the British Eedi online education platform, which collects mathematics learning data from secondary school students. What makes this dataset special is that it includes numerous mathematics problems with images, and each question is in multiple-choice format. The dataset contains over 15 million answer records, involving 120,000 students, 27,613 questions, and 388 different course topics.

4.2 Baselines

To validate the effectiveness of DDKT, we employed 9 different baselines. All models were trained and evaluated using default parameters on an A40 cluster. The details of all baselines are as follows:

- **DKT** is the first model to introduce deep learning into knowledge tracing[2], implemented based on Recurrent Neural Networks (RNN/LSTM). This model evaluates students' knowledge state by modeling their learning process and predicts their mastery of questions and corresponding knowledge concepts.
- **SAKT** is the first model to directly apply Transformer to knowledge tracing tasks, implemented based on self-attention mechanism[3]. This model proposes a self-attention model to capture long-term dependencies between student learning records, effectively improving the accuracy of knowledge state tracking.
- **sparseKT** is a knowledge tracing model based on the **SAKT** model[12]. Through k-sparse attention mechanism, it employs soft-thresholding and top-K sparsification strategies to select the most relevant historical interaction information, and introduces question discrimination factors to capture individual differences among different questions under the same knowledge point.
- **DTransformer** is a two-layer framework model based on Transformer, ensuring model stability through contrastive learning[26]. This model proposes a novel architecture to track students' learning activity patterns and improves model generalization through contrastive learning.
- **DKVMN** is a memory network-based knsowledge tracing model[10], implemented using a static key matrix and a dynamic value matrix. This model defines a key matrix to store latent knowledge concepts and a value matrix to store student knowledge state, updating student knowledge state through read and write operations.
- **DIMKT** is a knowledge tracing model that considers question difficulty effects, implemented based on deep learning architecture[15]. This model innovatively integrates question difficulty into the learning process, establishing relationships between student knowledge state and question difficulty levels, significantly improving prediction performance.
- **ReKT** is a simple yet powerful knowledge tracing model, implemented based on the FRU (Forget-Retrieve-Update) architecture[27]. This model models student knowledge state from multiple perspectives, including question, concept, and domain knowledge state, achieving excellent performance while maintaining simplicity.

- **extraKT** is a knowledge tracing model that extends attention model context windows through length extrapolation[28]. This model effectively addresses the limitations of attention mechanisms in processing long sequences, maintaining stable performance even with varying context window sizes.
- **simpleKT** is a simple yet powerful knowledge tracing baseline model, implemented based on ordinary dot-product attention function[29]. Inspired by the Rasch model in psychometrics, this model captures individual differences among questions with the same knowledge concepts through explicit modeling of question-specific variations, achieving excellent prediction performance while maintaining model simplicity.
- **stableKT** is an improved model based on **simpleKT**, enhancing length generalization ability by applying linear bias to attention scores[30]. This model uses multi-head aggregation modules to capture individual differences and complex hierarchical relationships, significantly improving prediction performance and length generalization ability.

4.3 Experiment Setup

In our experiments, we first deployed GLM-4 as the local LLM to extract question difficulty d_i^{llm}. The difficulty coefficient ranges from 0 to 100 as a percentage scale, where higher values indicate greater question difficulty. During the data loading phase, the statistical difficulty d_i^{stat} for question i is calculated for use in the Difficulty Perception Bias Sequence module.

Following the Pykt benchmark format[31], we divided each dataset into 5 folds, where each record represents a student's learning sequence. We used fold 0 as the test set and folds 1-4 as the training set. For model optimization, we employed the Adam optimizer with MSE Loss as the loss function. The learning rate was set to 0.0001, dropout rate to 0.2, and MLP embedding dimension to 256. The forgetting factor σ was set to 0.8 for calculating the Difficulty Mastery Ratio. In the multi-head attention mechanism, we set the layers with 4 attention heads. All experiments were conducted on the same A40 cluster to minimize hardware-related variations. In terms of model evaluation, we used AUC as the evaluation metric. AUC measures the model's ability to distinguish between correct and incorrect responses, and is a commonly used evaluation metric in knowledge tracing scenarios. Finally, we set early stop to 10 epochs to reduce training time.

4.4 Experiment Result

Table 2 demonstrates the performance comparison between our proposed DDKT model and baseline models across two datasets. The results show that our DDKT model consistently outperforms other baselines, particularly in capturing complex learning patterns and handling diverse educational scenarios. On large-scale datasets like XES3G5M, our DDKT model exhibits superior performance compared to other baseline models, achieving AUC improvements ranging from 2% to 7%. On smaller datasets such as Eedi, our DDKT model also achieves the best performance, with AUC improvements ranging from 1% to 8%. The comparison also reveals that our DDKT

Datasets / Models	XES3G5M			Eedi		
	AUC	Imp(abs.)	Imp(%)	AUC	Imp(abs.)	Imp(%)
DKT	83.02	4.44	5.35	79.41	1.62	2.04
SAKT	80.37	7.09	8.82	73.71	7.62	10.34
sparseKT	85.33	2.13	2.50	79.41	1.86	2.34
DTransform	84.97	2.49	2.93	79.17	1.60	2.02
DKVMN	82.47	4.99	6.05	76.04	4.99	6.56
DIMKT	85.81	1.65	1.92	79.41	1.86	2.34
ReKT	85.91	1.55	1.80	79.49	1.54	1.94
extraKT	85.48	1.98	2.32	79.50	1.53	1.92
simpleKT	85.33	2.13	2.50	79.44	1.59	2.00
stableKT	85.45	2.01	2.35	79.28	1.75	2.21
DDKT(Ours)	**87.46**	-	-	**81.03**	-	-

Table 2 The comparison between our propose DDKT model and other 9 baselines.

model performs particularly well on sparse datasets, the specific reasons for which will be discussed in the ***Ablation Study*** section.

4.5 Ablation Study

In our ablation study, we investigate the impact of three major components of DDKT on model performance: the Difficulty Mastery Ratio module, the Difficulty Perception Bias Sequence, and the Transformer Layer used in feature combination.

- DDKT w/o DMR: This variant removes the personalized Difficulty Mastery Ratio component for tracking student learning status;
- DDKT w/o DPBS: This variant removes the Difficulty Perception Bias Sequence component that combines objective and subjective difficulty measures for tracking student learning status;
- DDKT w/o Transformer: This variant replaces the Transformer architecture with ordinary Multihead Attention for combining input embeddings to track student learning status;

Datasets / Models	XES3G5M		Eedi	
	AUC	ACC	AUC	ACC
DDKT(Ours)	0.8746	0.8562	0.8103	0.7374
DDKT w/o DMR	0.8492	0.8479	0.7922	0.7208
DDKT w/o DPBS	0.8508	0.8486	0.7943	0.7233
DDKT w/o Transformer	0.8495	0.8476	0.7923	0.7202

Table 3 Comparison of three DDKT variants, demonstrating that our proposed model achieves superior performance among these variations, showing the improvement compared to other variants.

We tested the changes in AUC and ACC performance after removing or switching these modules, as shown in Table 3. On small datasets like Eedi and large dataset like XES3G5M, the results show that removing each module leads to decreases in both AUC and ACC, but it's more obvious in small dataset. Therefore, each module is

essential, and they work closely together. Regarding DPBS, when we only use statistical difficulty estimation for difficulty assessment, the following aspects are affected: When LLM assessment difficulty is removed, some questions perform worse due to the cold start problem, directly caused by data sparsity leading to statistical assessment difficulty values becoming too large or too small; Meanwhile, without LLM assessment difficulty, the process of establishing calibrated difficulty scores is missing. The model would adjust weights based on LLM assessment difficulty to calibrate statistical assessment difficulty, and the calibrated difficulty score is a more accurate difficulty calibration value that directly affects model performance. Due to the support of LLM assessment difficulty, the calibrated difficulty score reduces the impact of overly high or low statistical evaluation difficulties on the model, which is also the main reason why this model performs better on sparse datasets.

For DMR, removing the DMR module leads to a lack of student personalization modeling, which directly results in the model's uncertainty about students' difficulty mastery during the learning process. For example, when a student consistently performs well on high-difficulty algebra problems but struggles with basic arithmetic, traditional models might overestimate or underestimate their overall mathematical ability. The DMR module can capture this nuanced performance pattern and provide a more accurate portrait of the student's knowledge state. The TransformerEncoder encompasses Multi-head Attention and its feed-forward layers, and adds dropout layers to prevent overfitting and avoid local optima. Therefore, using Transformer can not only accomplish Multi-head Attention work but also prevent model overfitting, and can effectively capture long-range dependencies in student learning sequences through its self-attention mechanism.

4.6 DDKT in cold-start scenarios

The cold-start problem is inherent in the field of knowledge tracing. When new questions are introduced into the time series, models struggle to fit these new items effectively due to the lack of historical student interaction records, leading to the cold-start problem in knowledge tracing. We evaluated DDKT's performance in cold-start scenarios using AUC as our evaluation metric. To simulate cold-start scenarios, we utilized the XES3G5M dataset, using 1%, 5%, 10%, 20%, 30%, 40%, and 50% of student records as training sets, with each student's learning sequence being proportionally reduced. We selected two comparison models: DIMKT, which showed the best performance in Experiment Results as our baseline, and sparseKT, which specializes in sparse datasets, as our comparative models. The experimental results are shown in Figure 3.

The experimental results demonstrate that our DDKT model performed remarkably well even with extremely limited data, maintaining outstanding performance across all datasets. Notably, when the dataset was reduced to 10% of its original size, our DDKT model showed improvements of 1.83% compared to DIMKT and 2.23% compared to sparseKT; when reduced to 20%, DDKT achieved approximately 2% improvement over both DIMKT and sparseKT models. DIMKT, which solely relies on statistical difficulty assessment as a question difficulty feature, suffered from performance degradation due to partial information masking in the source

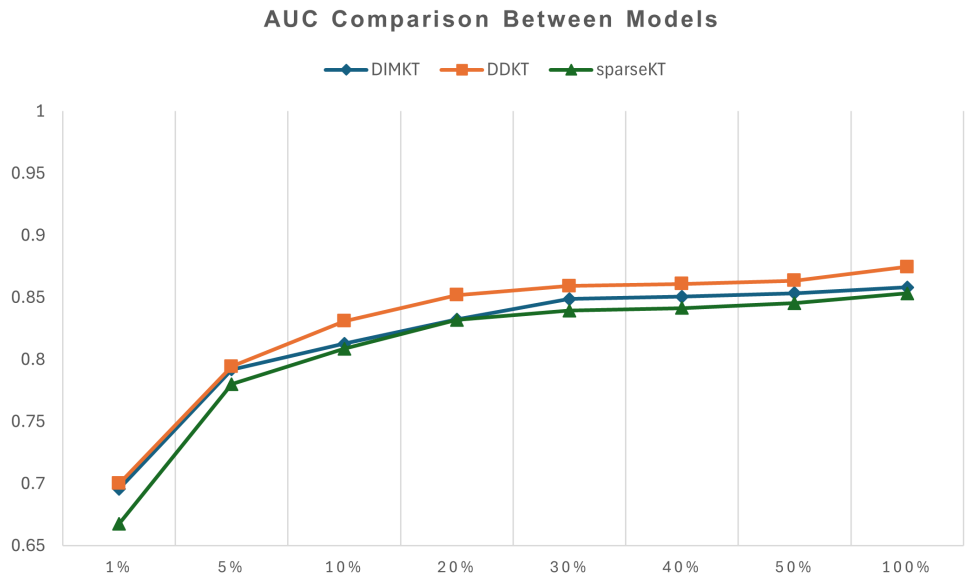

Fig. 3 Experimental results with cold start ratios of 1%, 5%, 10%, 20%, 30%, 40%, 50%, and 100%.

dataset, resulting in extremely high or low statistical difficulty assessments. Meanwhile, although sparseKT is designed to handle sparse datasets, its exclusive reliance on student learning records as input features limits its ability to fit well in data-scarce scenarios. Our proposed DDKT model incorporates LLM-evaluated difficulty coefficients as part of the input features and uses attention mechanisms to integrate them with statistical difficulty assessments, significantly reducing the impact of statistical difficulty assessment on the model and balancing cases where question difficulty features are either too high or too low.

4.7 Student Personalization with DDKT

In the process of learning gradually, there are inherent differences among students. Therefore, it is crucial for the model to accurately express the students' mastery of knowledge and learning progress in the process of knowledge tracking for personalized modeling. The Difficulty Mastery Ratio (DMR) we proposed reflects students' personalized learning characteristics by recording their performance on questions of varying difficulties, helping to construct a more precise representation of knowledge state. To validate the effectiveness of DMR in personalized modeling, we designed three comparative experiments, setting the students' Difficulty Mastery Ratios to all 0s, all 1s, and all random values, respectively. The experimental results are shown in Figure 4.

From the results, we found that when students' personalized difficulty mastery ratios were hidden, the model's performance was not as good, indicating that DMR indeed captured students' personalized characteristics during the learning process and significantly improved the model's predictive performance. When using all zeros,

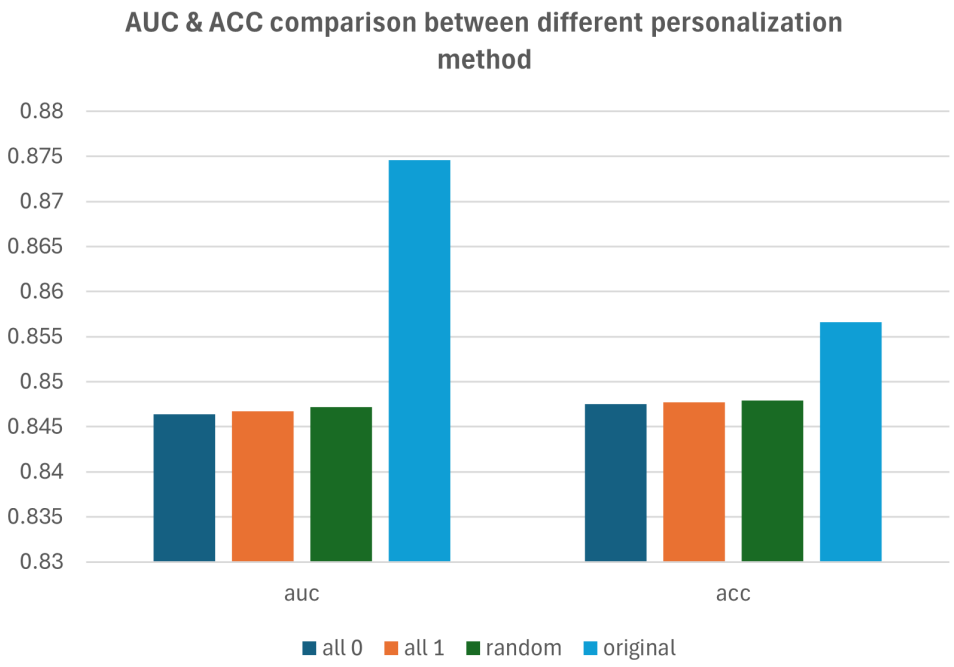

Fig. 4 The result of different personalization method.

all ones, or all random values, the AUC values were 84.64%, 84.67%, and 84.72% respectively. In comparison, our model achieved an AUC value of 87.46%, showing an improvement of approximately 3 percentage points over the baseline approaches. This significant performance improvement further validates the important role of DMR in personalized modeling. The experimental results demonstrate that accurately capturing and expressing students' personalized difficulty mastery characteristics has a crucial impact on the performance of knowledge tracing models, and our proposed DMR can effectively accomplish this task.

4.8 Visualizing Student's Study Process

To visualize the student's learning process more intuitively, we randomly selected one student's learning sequence in Eedi and recorded their dynamic difficulty adaptation indicators, knowledge state, difficulty perception bias sequence, and prediction probability values throughout their learning process, presenting them in the form of a heatmap. The sequence length is 199, representing the learning trajectory of 199 questions completed by the student. The heatmap of the student's learning process is shown in Figure 5.

During the initial 40 exercises, the objective difficulty of questions generally maintained a high level between 0.7-0.8, while the student's knowledge state fluctuated only between 0.4-0.5, with DPBS showing a positive bias of approximately 0.3-0.4 in

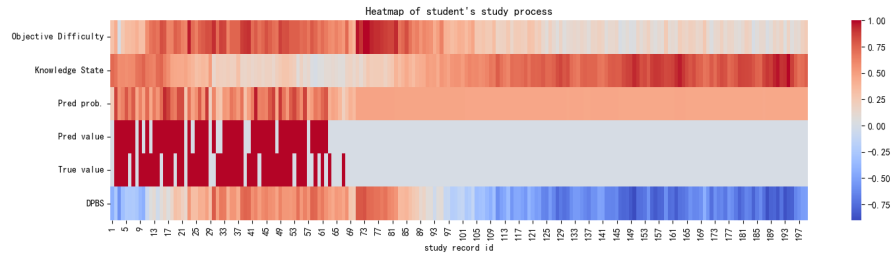

Fig. 5 We randomly selected a student's learning sequence and normalized the Objective as Difficulty and Knowledge State to a range between 0 and 1. Meanwhile, DPBS was calculated as Formula 3.

the heatmap. The prediction values during this phase were also relatively low, mostly maintaining around 0.5, reflecting the student's actual performance when facing challenging questions. In the subsequent transition period (exercises 40-120), we observed significant changes. The objective difficulty of questions gradually decreased to a moderate level of 0.5-0.6, while the student's knowledge state steadily increased to 0.6-0.7. The most crucial aspect during this phase was the change in DPBS, which gradually decreased from its initial positive value (around 0.3) to near zero, indicating that the student's ability level was gradually matching the question difficulty. The prediction values also rose to the 0.7-0.8 range, showing a clear improvement in learning effectiveness.

In the later learning stage (exercises 120-199), the student's performance reached a relatively stable level. The knowledge state consistently maintained a high level between 0.7-0.8, while question difficulty decreased to 0.3-0.4. At this point, the bias values stabilized in the negative range of -0.3 to -0.4, clearly indicating that the student's ability had significantly exceeded the question difficulty. The prediction values during this phase remained stable at a high level of 0.8-0.9, reflecting that the student had fully mastered this level of knowledge content. Through specific data changes, the model effectively simulated the gradual nature of the complex "learning" process. From the initial gap between objective difficulty and knowledge state (0.3-0.4) to the final achievement of stable negative bias (-0.3 to -0.4), it demonstrated the quantifiable progress in the student's learning process.

5 Conclusion

Difficulty, as a crucial indicator in the learning process, deserves thorough investigation in its semantic data mining and application in knowledge tracing. Previous difficulty models typically employed single difficulty indicators, either using LLM-assessed difficulty or statistical difficulty, without combining subjective question difficulty with students' objective difficulty. To deeply explore the impact of difficulty features on knowledge tracing and mitigate the cold start problem common in knowledge tracing, our proposed DDKT model incorporates LLM-assessed difficulty as part of the calibrated difficulty fusion, combines students' subjective and objective difficulty

perceptions, and achieves more precise student modeling. The combination of LLM-assessed and statistical difficulty provides educators with enhanced interpretability by allowing them to analyze learning patterns from both algorithmic and statistical perspectives, enabling more informed decision-making in educational interventions and content adaptation. Our DDKT model demonstrates superior performance among all baselines, and ablation studies confirm the collaborative effectiveness of its three modules. In our future work, we will focus on exploring deeper connections between student performance and difficulty from the student perspective, such as the relationship between problem-solving time and difficulty, the connection between repeated attempts and difficulty, etc., to construct a more refined knowledge tracing model that improves both accuracy and interpretability.

Declarations

Competing interests

The authors declared no potential conflicts of interest with respect to the re-search, authorship, and/or publication of this article.

Data, Materials and/or Code availability

The data are available from the corresponding author on reasonable request.

Funding

This work was supported by funding from Guangdong Philosophy and Social Science Foundation (Grant No. GD22WZX02-03), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515110134) and Guangzhou Municipal Science and Technology Program (Grant No. 2024A04J3752).

References

[1] Abdelrahman, G., Wang, Q., Nunes, B.: Knowledge tracing: A survey. ACM Computing Surveys **55**(11), 1–37 (2023)

[2] Piech, C., Bassen, J., Huang, J., Ganguli, S., Sahami, M., Guibas, L.J., Sohl-Dickstein, J.: Deep knowledge tracing. Advances in neural information processing systems **28** (2015)

[3] Pandey, S., Karypis, G.: A self-attentive model for knowledge tracing. arXiv preprint arXiv:1907.06837 (2019)

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

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

[6] Tong, H., Wang, Z., Zhou, Y., Tong, S., Han, W., Liu, Q.: Hgkt: Introducing hierarchical exercise graph for knowledge tracing. arXiv preprint arXiv:2006.16915 (2020)

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

[8] Yeung, C.-K.: Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. arXiv preprint arXiv:1904.11738 (2019)

[9] Chen, J., Liu, Z., Huang, S., Liu, Q., Luo, W.: Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 14196–14204 (2023)

[10] Zhang, J., Shi, X., King, I., Yeung, D.-Y.: Dynamic key-value memory networks for knowledge tracing. In: Proceedings of the 26th International Conference on World Wide Web, pp. 765–774 (2017)

[11] Liu, Z., Liu, Q., Chen, J., Huang, S., Gao, B., Luo, W., Weng, J.: Enhancing deep knowledge tracing with auxiliary tasks. In: Proceedings of the ACM Web Conference 2023, pp. 4178–4187 (2023)

[12] Huang, S., Liu, Z., Zhao, X., Luo, W., Weng, J.: Towards robust knowledge tracing models via k-sparse attention. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2441–2445 (2023)

[13] Pardos, Z.A., Heffernan, N.T.: 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 19, pp. 243–254 (2011). Springer

[14] Wang, Z., Feng, X., Tang, J., Huang, G.Y., Liu, Z.: Deep knowledge tracing with side information. In: Artificial Intelligence in Education: 20th International Conference, AIED 2019, Chicago, IL, USA, June 25-29, 2019, Proceedings, Part II 20, pp. 303–308 (2019). Springer

[15] Shen, S., Huang, Z., Liu, Q., Su, Y., Wang, S., Chen, E.: Assessing student's dynamic knowledge state by exploring the question difficulty effect. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 427–437 (2022)

[16] Lee, U., Yoon, S., Yun, J.S., Park, K., Jung, Y., Stratton, D., Kim, H.: Difficulty-focused contrastive learning for knowledge tracing with a large language model-based difficulty prediction. arXiv preprint arXiv:2312.11890 (2023)

[17] Zhang, F., Qiu, L., Cui, Z., Chen, X.: A difficulty personalization model based on heterogeneous graph for knowledge tracing. Available at SSRN 5009691

[18] Qiu, L., Wang, L.: Knowledge tracing through enhanced questions and directed learning interaction based on multigraph embeddings in intelligent tutoring systems. IEEE Transactions on Education (2024)

[19] Tan, M., Merrill, M.A., Gupta, V., Althoff, T., Hartvigsen, T.: Are language models actually useful for time series forecasting? In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024)

[20] Yu, Y., Zhou, Y., Zhu, Y., Ye, Y., Chen, L., Chen, M.: Eckt: Enhancing code knowledge tracing via large language models. In: Proceedings of the Annual Meeting of the Cognitive Science Society, vol. 46 (2024)

[21] Guo, Y., Shen, S., Liu, Q., Huang, Z., Zhu, L., Su, Y., Chen, E.: Mitigating cold-start problems in knowledge tracing with large language models: An attribute-aware approach. In: Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, pp. 727–736 (2024)

[22] Yang, L., Sun, X., Li, H., Xu, R., Wei, X.: Dpkt: Difficulty-aware programming knowledge tracing with large language models (2024)

[23] Ciniselli, M., Cooper, N., Pascarella, L., Poshyvanyk, D., Di Penta, M., Bavota, G.: An empirical study on the usage of bert models for code completion. In: 2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR), pp. 108–119 (2021). IEEE

[24] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q.V., Zhou, D., *et al.*: Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems **35**, 24824–24837 (2022)

[25] Liu, Z., Liu, Q., Guo, T., Chen, J., Huang, S., Zhao, X., Tang, J., Luo, W., Weng, J.: Xes3g5m: A knowledge tracing benchmark dataset with auxiliary information. Advances in Neural Information Processing Systems **36** (2024)

[26] Yin, Y., Dai, L., Huang, Z., Shen, S., Wang, F., Liu, Q., Chen, E., Li, X.: Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer. In: Proceedings of the ACM Web Conference 2023, pp. 855–864 (2023)

[27] Shen, X., Yu, F., Liu, Y., Liang, R., Wan, Q., Yang, K., Sun, J.: Revisiting knowledge tracing: A simple and powerful model. In: Proceedings of the 32nd

ACM International Conference on Multimedia, pp. 263–272 (2024)

[28] Li, X., Bai, Y., Guo, T., Zheng, Y., Hou, M., Zhan, B., Huang, Y., Liu, Z., Gao, B., Luo, W.: Extending context window of attention based knowledge tracing models via length extrapolation. In: ECAI 2024, pp. 1479–1486. IOS Press, ??? (2024)

[29] Liu, Z., Liu, Q., Chen, J., Huang, S., Luo, W.: simplekt: a simple but tough-to-beat baseline for knowledge tracing. arXiv preprint arXiv:2302.06881 (2023)

[30] Li, X., Bai, Y., Guo, T., Liu, Z., Huang, Y., Zhao, X., Xia, F., Luo, W., Weng, J.: Enhancing length generalization for attention based knowledge tracing models with linear biases. In: 33rd International Joint Conference on Artificial Intelligence, IJCAI 2024, pp. 5918–5926 (2024). International Joint Conferences on Artificial Intelligence

[31] Liu, Z., Liu, Q., Chen, J., Huang, S., Tang, J., Luo, W.: pykt: a python library to benchmark deep learning based knowledge tracing models. Advances in Neural Information Processing Systems **35**, 18542–18555 (2022)