

Disentangling Knowledge States with Ability and Proficiency Modeling for Knowledge Tracing

Duantengchuan Li[✉], Yingqian Bi[✉], Jinsong Chen[✉], Rui Zhang[✉], Mingwen Tong[✉]

Abstract—Knowledge tracing (KT) aims to predict students’ future performance by modeling their evolving knowledge states from historical interactions. Existing KT methods usually treat the raw interaction sequence as a unified behavioral process, overlooking the phase-specific nature of learning behaviors. Our preliminary observations show that students are more likely to correctly answer previously failed knowledge concepts after sufficient practice, suggesting a transition from ability-building to proficiency-oriented learning. Motivated by this, we propose Phase-Aware Knowledge Tracing (PAKT), a KT framework that decomposes student interactions into ability and proficiency phases based on the tailored decomposition mechanism. To effectively exploit the decomposed sequences, we design a multi-branch Transformer with a type-aware readout module to jointly capture phase-specific and holistic knowledge states. We further provide a causal analysis to reveal the confounding bias caused by entangling complex learning behaviors in phase-agnostic KT models. Extensive experiments on six public benchmarks demonstrate that our method consistently outperforms representative baselines, with a maximum AUC gain of 1.33% and an average gain of 0.82%.

Index Terms—Knowledge Tracing, Ability Phase, Proficiency Phase, Decomposed Sequences, Causal Analysis

I. INTRODUCTION

With the development of educational data mining and learning analytics technologies, Intelligent Tutoring Systems (ITS) have gained significant attention[1]. An ITS is a computer-based system that delivers personalized instruction and adaptive feedback to students. By modeling the strengths and weaknesses of a student, ITS provides adaptive problem selection, learning performance prediction, and timely feedback to facilitate effective learning outcomes. A central challenge of ITS lies in accurately modeling the evolving knowledge state of individual students. To address this, Knowledge Tracing (KT), which aims to estimate a student’s mastery of knowledge concepts from their historical interaction sequences, has emerged as a fundamental task in ITS.

Early KT models are grounded in probabilistic modeling psychometric theory, including Hidden Markov Models (HMMs) and Item Response Theory (IRT), for student performance prediction [2, 3]. DKT[4] first introduces deep learning

to the KT task with recurrent neural networks (RNNs), substantially outperforming prior probabilistic approaches. Since each student’s interaction history is naturally sequential, deep learning architectures for sequential data have been widely applied to the KT task [5–7]. Sequential KT research has converged on three primary modeling approaches. The first consists of deep sequential models, such as the RNN-based DKT[4] and the Transformer-based SAINT [8, 9], which capture temporal dependencies in student interaction sequences to estimate evolving knowledge states. The second covers structure-enhanced KT models, which incorporate auxiliary structures, such as external memory in DKVMN[10] and concept prerequisite graphs in GKT[11] and GRKT[12]. The third comprises interpretable KT models, such as EKT[13], XKT [14, 15], and DisenKT[16], which enhance transparency by incorporating educational theories or learning disentangled representations of student knowledge states [17–20].

Despite the remarkable progress achieved by existing KT methods [21, 22], most of them directly model the raw student interaction sequence with neural networks, implicitly assuming that all interactions contribute to the evolution of students’ knowledge states in a same manner [23, 24]. However, such a unified modeling paradigm overlooks the fact that students may exhibit substantially different learning behaviors at different phases of knowledge acquisition. A recent study DisKT[25] decomposes students’ response sequences based on response correctness, and learns separate representations of familiar and unfamiliar abilities from the respective sub-sequences. However, this correctness-based decomposition is too coarse to represent the behavioral semantics of different learning phases, such as initial knowledge acquisition versus skill consolidation.

In practice, a student’s response behavior before and after mastering a knowledge concept can be quite different. Early interactions mainly reflect the learner’s ability development through repeated practice, while later interactions better indicate the stability and utilization of mastered knowledge. As shown in Fig. 1, a substantial proportion of students are more likely to answer knowledge concepts correctly in later interactions after sufficient practice. Therefore, directly encoding the entire interaction sequence without distinguishing these phases may mix complex learning signals and limit the model’s ability to precisely characterize the dynamic evolution of learners’ knowledge phases. This phenomenon raises a challenging question: *How can we effectively model phase-specific learning behaviors from raw student interaction sequences to better characterize the dynamic evolution of learners’ knowledge states?*

Duantengchuan Li and Rui Zhang are with the School of the Information Management, Wuhan University, Wuhan 430072, China.

Yingqian Bi is with the School of Computer Science and Artificial Intelligence, Hubei University of Technology, Wuhan 430068, China; School of the Information Management, Wuhan University, Wuhan 430072, China

Jinsong Chen and Mingwen Tong are with the Faculty of Artificial Intelligence in Education, Central China Normal University, Wuhan 430074, China.

The first two authors contribute equally.

(Corresponding author: Jinsong Chen and Rui Zhang. E-mail: guangni-anchenai@ccnu.edu.cn and ruizhang8633@gmail.com.)

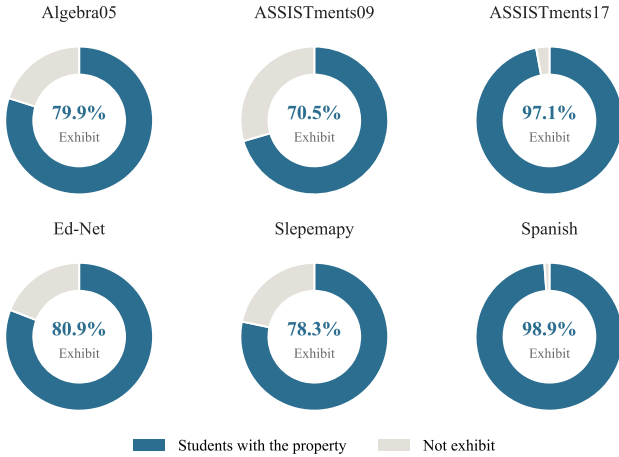


Fig. 1: Proportion of students exhibiting the two-phase learning property across six benchmark datasets. A student is considered to exhibit the property if their late-phase accuracy exceeds early-phase accuracy on more than a quarter of qualifying knowledge concepts.

To this end, we propose Phase-Aware Knowledge Tracing (PAKT), a new KT model that decomposes the student interaction sequence into ability and proficiency phases for phase-specific modeling. The learning process comprises two phases. In the initial phase, students gradually master knowledge concepts through practice. Once mastery is achieved, further practice drives skill automation, leading to more fluent and consistent performance rather than additional gains in correctness. Accordingly, our decomposition mechanism partitions each learner’s response history based on the cumulative correct response count per knowledge concept, assigning early interactions to the ability phase and later ones to the proficiency phase. The two decomposed subsequences, together with the original complete sequence, are fed into a multi-branch Transformer-based backbone, where separate decoders encode the ability, proficiency, and holistic aspects of the student’s knowledge state. A type-aware readout module then integrates these complementary representations into a unified knowledge state for final prediction. The contributions of this paper are summarized as follows:

- We develop a new learning behavior-aware decomposition mechanism that partitions student interaction sequences into ability and proficiency phases.
- We propose a multi-branch Transformer-based backbone with a type-aware readout module to model the decomposed interaction sequences for KT tasks.
- Extensive experiments on six public benchmarks demonstrate that PAKT achieves consistent improvements over all representative baselines.

II. RELATED WORK

In this section, we first provide a comprehensive review of recent efforts in KT. Then, we introduce the approaches of applying sequence decomposition mechanism in the KT task.

A. Knowledge Tracing

Prior approaches for KT tasks are built on probabilistic models and psychometric theory. BKT[2], a classic probabilistic model, employs a hidden Markov model to simulate a student’s knowledge mastery and makes predictions based on their historical exercise performance. TLS-BKT[26] applied stage split based on BKT, but it still remains within the probabilistic graphical modeling framework and does not leverage the representational power of deep neural networks. Item Response Theory (IRT)[3], a psychometric model, estimates the probability of a correct response by modeling the interaction between a person’s latent ability (e.g., knowledge level) and item characteristics such as difficulty and discrimination. However, these methods rely on strong assumptions and manually defined knowledge structures, which restricts their applications in the real-world scenario[27].

Recently, deep learning-based approaches for KT tasks have attracted great attention due to the superior performance and flexibility. DKT[4] usher in a new era for KT by introducing RNNs to directly model sequential student interactions. Then, KT models try to represent knowledge states more explicitly by introducing more complex neural networks. DKVMN[10] is proposed with a memory-augmented network with separate matrices for concepts and mastery states, while AKT[28] and SAINT[8] leverage Transformer[9] attention to dynamically weigh historical interactions.

Building upon these advancements in sequential modeling, subsequent research explore ways to incorporate external knowledge structures and novel training paradigms to further enhance model capability. Besides sequential models, advanced techniques in deep learning have been introduced into KT. GKT[11], a graph-based KT model, operates on a prerequisite graph using Graph Neural Networks (GNNs) to capture dependencies between knowledge concepts. To enhance model robustness, CL4KT[29] employs contrastive learning to derive noise-invariant student representations, and DTransformer[30] integrates diagnostic decoders to isolate concept-specific mastery levels.

To enhance interpretability, educational theories have been integrated into deep KT models. Deep-IRT[5] integrates the classic Item Response Theory (IRT) into a deep learning framework, allowing the model’s parameters (e.g., ability, difficulty) to retain their psychometric meanings. MIKT[19] explicitly models the monotonic relationship between a student’s knowledge state and their probability of answering correctly, a core principle in education that ensures interpretable predictions. CoreKT[31] focuses on concept-level interpretability by designing a model that guarantees monotonicity at the level of individual knowledge components, making its predictions more transparent and educationally plausible.

In parallel, a contrasting line of work seeks simplicity and efficiency. SparseKT[32] tackles data sparsity in long sequences via a sparse Transformer architecture, replacing full attention with localized patterns to improve efficiency and generalization. SimpleKT[22] advocates for model simplicity, showing that a well-regularized, single-layer LSTM can match the performance of far more complex KT models.

In summary, prior KT models largely adopt a homogeneous modeling paradigm, applying a single system to the entire learning sequence. This approach inherently conflates the distinct dynamics of early knowledge acquisition (driven by cognitive ability) and later skill consolidation (driven by practice). Our work breaks from this tradition by introducing a temporally-grounded, data-driven decomposition of the sequence, enabling explicit and separate modeling of these two fundamental learning phases.

B. Decomposition Mechanism in Knowledge Tracing

Introducing sequence decomposition into KT tasks is a promising way to capture detailed features in learning process. DisKT [25] pioneers the application of sequence decomposition in KT by employing structural causal model to deconfound spurious correlations and estimate counterfactual effects.

However, its intervention mechanism relies on a simplistic partitioning of student interactions (e.g., by historical performance) and fails to model the distinct educational semantics, such as the phase of initial knowledge acquisition versus skill consolidation.

In contrast, the proposed PAKT introduces a tailored decomposition strategy that divides the history records into ability phase and proficiency phase, facilitating the causal inference-based technique deeply modeling the learning behaviors.

III. PRELIMINARIES

A. Problem Formulation

The KT task aims to model the evolution of a student's latent knowledge state based on their observed learning interactions. We begin by defining the core elements of an educational system:

- $\mathcal{S} = \{s_1, s_2, \dots\}$: the set of all students.
- $\mathcal{Q} = \{q_1, q_2, \dots\}$: the set of all questions.
- $\mathcal{C} = \{c_1, c_2, \dots\}$: the set of all knowledge concepts (or skills).

For a given student $s \in \mathcal{S}$, their learning process over T discrete time steps is recorded as a sequential interaction history:

$$X^{(s)} = \{x_1, x_2, \dots, x_T\}.$$

Each interaction x_t at time step t is represented as a triple:

$$x_t = (q_t, \mathcal{C}_{q_t}, r_t),$$

where:

- $q_t \in \mathcal{Q}$ denotes the question answered at step t .
- $\mathcal{C}_{q_t} \subseteq \mathcal{C}$ is the set of knowledge concepts assessed by question q_t , i.e., $\mathcal{C}_t = \{c_t^{(1)}, c_t^{(2)}, \dots, c_t^{(p)}\}$.
- $r_t \in \{0, 1\}$ is the binary response label, with $r_t = 1$ indicating a correct answer and $r_t = 0$ indicating an incorrect one.

Formally, the KT task is defined as follows: given a student's historical interaction sequence $X^{(s)}$ up to time T , and the next question q_{T+1} (along with its associated concepts $\mathcal{C}_{T+1}$), the

goal is to predict the probability that the student will answer q_{T+1} correctly:

$$P(r_{T+1} = 1 \mid X^{(s)}, q_{T+1}, \mathcal{C}_{q_{T+1}}).$$

This probability reflects the model's estimation of the student's mastery of concepts in $\mathcal{C}_{T+1}$ at the moment of prediction, conditioned on their entire learning trajectory.

IV. METHODOLOGY

In this section, we detail our proposed PAKT, which includes three core components: learning behavior-aware sequence decomposition, multi-branch Transformer-based backbone and type-aware readout module. These components are utilized for generating input sequences, learning representation and producing the final representations for prediction, respectively. After introducing the above components, we provide the theoretical analysis of PAKT from the perspective of causal inference.

A. Learning Behavior-Aware Sequence Decomposition

During the initial learning of a knowledge concept (KC), students first aim to master the core concept and practice it repeatedly to master its application across different contexts. The repeated practice increases their familiarity, thereby raising the probability of a correct response involving the specific KC. Subsequently, further practice leads to skill automation [33], as students have internalized the common solution patterns for this KC. This illustrates that proficiency is dynamically built up over time. In summary, the dominant factor influencing a student's performance undergoes a shift: from initial knowledge acquisition to the consolidation and automation of skills through practice. This situation motivates us to decompose the raw interaction sequence into separate subsequences containing different learning phases to deeply understand the learning behaviors of the target student.

Specifically, we decompose the learning process into two phases, the ability phase and the proficiency phase:

- **Ability Phase:** This initial phase captures the period when a student encounters a new knowledge concept and is actively working to understand and master its core principles. During this phase, the student's performance is primarily governed by their intrinsic cognitive aptitude for grasping the concept.
- **Proficiency Phase:** At this stage, the core knowledge is assumed to be acquired, and performance is increasingly driven by the effects of practice, leading to skill automation and fluency. The student's probability of a correct answer becomes more influenced by the recency, frequency, and variety of practice rather than initial aptitude. This phase is therefore optimal for modeling the dynamic, practice-induced factor of proficiency, which captures how well a student can reliably and efficiently apply the knowledge across different problem contexts.

Based on the descriptions of the above two phases in the learning process, we now detail the generation mechanism of interaction sequences for each phase. Given the initial interaction sequence of the student s , we first transform the raw

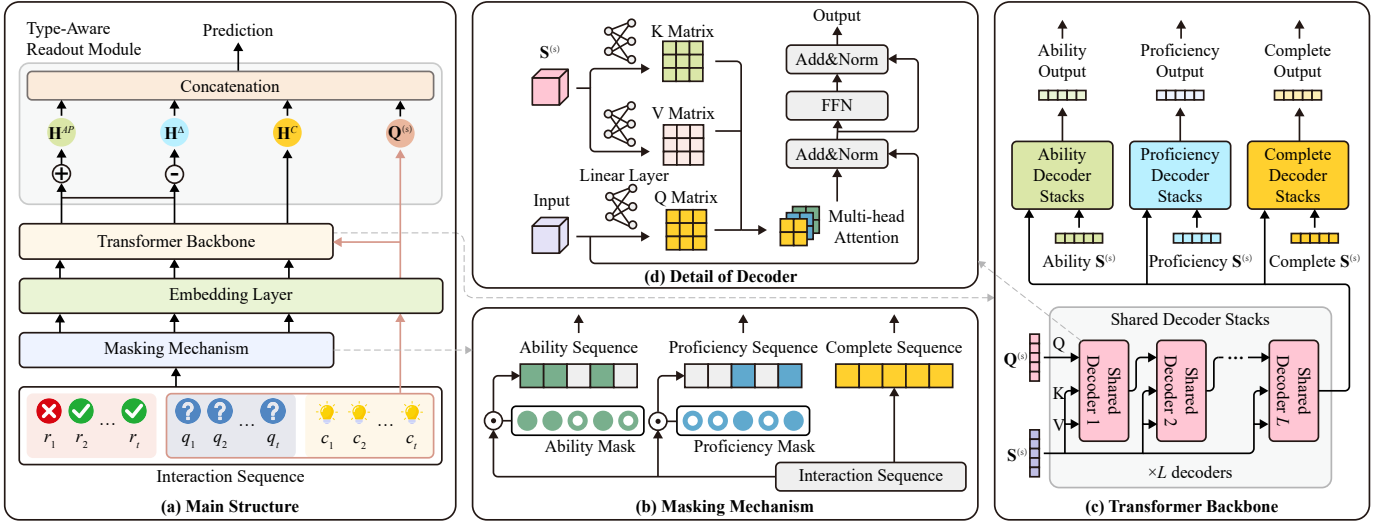


Fig. 2: An overview of the PAKT model. The response correctness, question ID, and concept ID sequences are processed through the mask mechanism and Embedding Layer into the Transformer-based Backbone. The highlighted path (red box and arrow) in Fig. 2 (a) indicates that the question ID and concept ID are fed solely into the embedding layer, whose output directly serves as the question feature representations $\mathbf{Q}^{(s)}$. The three-branch outputs of the Backbone, $\mathbf{H}^A$, $\mathbf{H}^P$, and $\mathbf{H}^C$, together with $\mathbf{Q}^{(s)}$, are subsequently aggregated by the Type-Aware Readout Module to produce the final prediction.

response sequence into the KC-based interaction sequences according to the previous study [22, 25]. Then, for each KC c at the t -th step in the KC sequence, we define a binary *phase indicator function* $\delta(s, c, t)$ based on a decomposing threshold k and the cumulative correct attempts:

$$\delta(s, c, t) = \mathbb{I}(\gamma(s, c, t) < k), \quad (1)$$

where $\gamma(s, c, t)$ denotes the number of times student s has correctly answered questions related to concept c at the t -th step, and k is a hyper-parameter. $\mathbb{I}(\cdot) \in \{0, 1\}$ denotes the indicator function.

According to the output of the phase indicator function $\delta(s, c, t)$, we further develop two decomposition mask matrices $\mathbf{M}^{(s),A} \in \mathbb{R}^{L_s^{KC} \times C}$ and $\mathbf{M}^{(s),F} \in \mathbb{R}^{L_s^{KC} \times C}$ to achieve the learning behavior-aware sequence decomposition, where L_s^{KC} denotes the length of the student's KC sequence, $\mathbf{M}^{(s),A}$ and $\mathbf{M}^{(s),F}$ are adopted for the ability phase and the proficiency phase, respectively. Specifically, for the given interaction KC tuple (c, t) , the corresponding values in $\mathbf{M}^{(s),A}$ and $\mathbf{M}^{(s),F}$ are defined as follows:

$$\mathbf{M}_t^{(s),A} = \delta(s, c, t), \quad \mathbf{M}_t^{(s),F} = 1 - \delta(s, c, t). \quad (2)$$

Based on the Eq. (2), an interaction at time t involving concept c is assigned to the proficiency phase if and only if the cumulative correct count for that concept has reached or exceeded the threshold k . Otherwise, this KC tuple will be assigned to the ability phase. This binary decomposing ensures that each interaction for a given concept exclusively belongs to one of the two phases, thereby enabling separate modeling of the ability and proficiency components throughout the learning sequence. For simplicity, we consider only the scenario where each item contains one knowledge concept. Having multiple knowledge concepts does not impact the

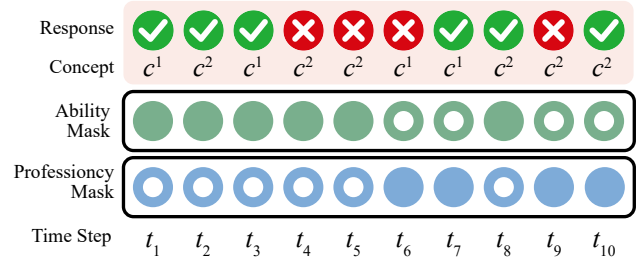


Fig. 3: The illustration of the decomposition mask generation. Taking the cumulative correct answer threshold $k = 2$ as the dividing criterion: for concept c^1 and c^2 , the cumulative correct responses first reach twice at time step t_3 and t_8 , respectively. Thus the interactions at t_1, t_3 and t_8 are divided into the ability phase, while the interactions at t_6, t_7, t_9 and t_{10} belong to the proficiency phase.

method's implementation. Fig.3 illustrates the generation of two masks.

B. Multi-Branch Transformer-Based Backbone

After constructing the decomposition mask matrices, PAKT further develops a multi-branch Transformer-based backbone to learn the knowledge states of student s from the input interaction sequences via the generated decomposition masks $\mathbf{M}^{(s),A}$ and $\mathbf{M}^{(s),F}$.

Based on previous approaches[28][30][25], we first construct the initial input features of the KC-based interaction sequences for the student s . Specifically, the initial representations of t -th interaction (c_t, r_t) are calculated as follows:

$$\mathbf{Q}_t^{(s)} = \mathbf{c}_{c_t} + \mathbf{d}_{q_t}, \quad \mathbf{S}_t^{(s)} = \mathbf{c}_{c_t} + \mathbf{r}_{r_t} + \mathbf{d}_{q_t}, \quad (3)$$

where $\mathbf{c}_{c_t} \in \mathbb{R}^d$ is the embedding of concept c_t , $\mathbf{d}_{q_t} \in \mathbb{R}^d$ is a learnable vector representing the difficulty of question q_t and $c_t \in \mathcal{C}_{q_t}$. $\mathbf{r}_{r_t} \in \mathbb{R}^d$ is the embedding of response r_t . d denotes the dimension of these embeddings. $\mathbf{Q}^{(s)} \in \mathbb{R}^{L_s^{KC} \times d}$ and $\mathbf{S}^{(s)} \in \mathbb{R}^{L_s^{KC} \times d}$ are the initial representations of the corresponding KC c_t and the interaction tuple (c_t, r_t) .

Then, PAKT adopts the generated decomposition masks to create three types of input branches for comprehensively capturing the semantic information from the student's learning behaviors:

$$\mathbf{Q}^A = \mathbf{Q}^{(s)} \odot \mathbf{M}^{(s),A}, \mathbf{Q}^F = \mathbf{Q}^{(s)} \odot \mathbf{M}^{(s),F}, \mathbf{Q}^C = \mathbf{Q}^{(s)}, \quad (4)$$

$$\mathbf{S}^A = \mathbf{S}^{(s)} \odot \mathbf{M}^{(s),A}, \mathbf{S}^F = \mathbf{S}^{(s)} \odot \mathbf{M}^{(s),F}, \mathbf{S}^C = \mathbf{S}^{(s)}. \quad (5)$$

The descriptions of the above three branches are as follows:

- $\mathbf{Q}^A$ and $\mathbf{S}^A$ are the ability branch input which processes the subset of interactions identified as belonging to the *ability phase* by the ability mask $\mathbf{M}^{(s),A}$. This sequence primarily reflects the student's intrinsic cognitive capacity for initial knowledge acquisition. The corresponding ability representation $\mathbf{H}^A$ is learned from this branch.
- $\mathbf{Q}^F$ and $\mathbf{S}^F$ are the proficiency branch input which operates on the complementary subset marked by the proficiency mask $\mathbf{M}^{(s),F}$, corresponding to the *proficiency phase*. This sequence captures the dynamics of practice consolidation, skill automation, and performance consistency after initial mastery. The proficiency representation $\mathbf{H}^F$ is derived from this branch.
- $\mathbf{Q}^C$ and $\mathbf{S}^C$ are the complete branch which processes the original, non-decomposed interaction sequence. It serves to maintain a holistic view of the student's overall learning trajectory. The comprehensive student representation $\mathbf{H}^C$ is obtained from this branch.

In practice, we also adopt the response-aware strategy introduced by [25] to strengthen the interaction decomposition. PAKT further leverages a two-stage Transformer-based backbone to learn the dynamic knowledge states from the input three branches. Specifically, in the first stage, all input branches share the same Transformer decoder. While, in the second stage, each branch is fed to independent Transformer decoder. The shared first stage learns universal sequential features for efficiency, and dedicated second-stage decoders enable phase-specific specialization for representation learning.

The forward pass of the first decoder layer in this shared stage is defined as:

$$\mathbf{H}_1^b = \text{Decoder}(Q = \mathbf{Q}^b, K = \mathbf{S}^b, V = \mathbf{S}^b), \quad (6)$$

where $b \in \{A, F, C\}$ represents the branch indicator. $\text{Decoder}(\cdot)$ represents the standard Transformer decoder which consists of L Transformer layers.

Similarly, the forward pass of the second decoder layer in phase-specific stage is defined as:

$$\mathbf{H}^b = \text{Decoder}(Q = \mathbf{H}_1^b, K = \mathbf{S}^b, V = \mathbf{S}^b), \quad (7)$$

where $\text{decode}^b(\cdot)$ represents the standard Transformer decode which consists of L Transformer layers for branch b .

C. Type-Aware Readout Module

After obtaining the three student representations (ability $\mathbf{H}^A$, proficiency $\mathbf{H}^F$, and complete $\mathbf{H}^C$) from the Transformer-based backbone, we design a fusion mechanism to integrate them into a comprehensive student representation.

First, the ability and proficiency representations are adaptively fused via a learnable coefficient $\alpha \in [0, 1]$:

$$\mathbf{H}^{AF} = \alpha \cdot \mathbf{H}^A + (1 - \alpha) \cdot \mathbf{H}^F, \quad (8)$$

where α is optimized during training to automatically balance the contribution of ability and proficiency.

To capture the differential gain from practice, especially for students who improve significantly after the initial learning phase, we compute the proficiency-ability gap:

$$\mathbf{H}^\Delta = \mathbf{H}^F - \mathbf{H}^A, \quad (9)$$

This representation encodes the extent to which practice has enhanced performance beyond the student's initial ability.

The final comprehensive student representation $\mathbf{H}^S$ is then constructed by concatenating the fused representation $\mathbf{H}^{AF}$, the complete-branch representation $\mathbf{H}^C$, and the differential gain vector $\mathbf{H}^\Delta$:

$$\mathbf{H}^S = [\mathbf{H}^{AF} \parallel \mathbf{H}^C \parallel \mathbf{H}^\Delta]. \quad (10)$$

where $[\cdot \parallel \cdot]$ denotes vector concatenation. This design ensures the model retains information from the holistic learning trajectory while preserves the semantic behavioral features from different learning phases.

D. Response Prediction and Model Training

The prediction layer takes the concatenated joint representation $\mathbf{z} = [\mathbf{H}^S \parallel \mathbf{Q}^{(s)}] \in \mathbb{R}^{4d}$ which combines the representations of the student and the target question as input and outputs the predicted probability of a correct response. This layer is implemented as a multi-layer perceptron (MLP) with non-linear activation:

$$\hat{y} = \sigma(\text{MLP}(\mathbf{z})), \quad (11)$$

where $\sigma(\cdot)$ denotes the sigmoid activation function, $\hat{y} \in (0, 1)$ denotes the predicted probability that the student will answer the question correctly.

The model is trained using a composite loss function that combines two objectives. The primary component is the binary cross-entropy loss between predicted and actual response outcomes:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)], \quad (12)$$

where $y_i \in \{0, 1\}$ represents the ground truth correctness label, $\hat{y}_i \in [0, 1]$ is the model's predicted probability, and N is the number of valid training samples.

To accelerate the model convergence and explicitly guide the ability branch to learn distinct skill learning patterns, we introduce a regularization term applied specifically to this branch:

$$\mathcal{L}_{\text{reg}} = -\frac{1}{U} \sum_{u=1}^U \|\mathbf{S}_u^{A+} - \mathbf{S}_u^{A-}\|_2, \quad (13)$$

where S_u^{A+} and S_u^{A-} represent the processed embeddings for positive (correct) and negative (incorrect) samples for ability branch within batch u , respectively.

The final loss function combines the binary cross-entropy loss with the regularization term:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{reg}}. \quad (14)$$

E. Causality Analysis on PAKT

To deeply understand the proposed PAKT, we provide the causality analysis [34, 35] on PAKT. Specifically, we construct a structural causal model that identifies the factors influencing student performance. As shown in Fig. 4, the colliding effect between a student's intrinsic ability and acquired proficiency on student features introduces confounding bias, a key limitation conflating these factors in traditional KT models.

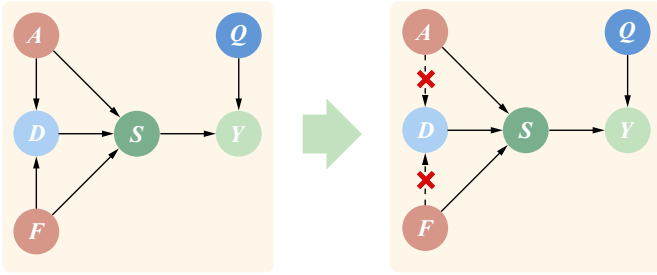


Fig. 4: The causal graph of conventional KT model and DAPKT, where the direct causal paths from ability (A) and proficiency (F) to student features (S) are explicitly cut.

1) *Causal Graph*: In the structural causal graph, we define the following key variables and their causal relationships:

- A denotes the student's cognitive ability.
- F denotes the student's proficiency on knowledge concepts.
- D denotes the distribution of the student's historical interaction sequence.
- S denotes latent vector that encodes the student's features.
- Y denotes the predicted probability of the student answering a given question correctly.
- Q denotes feature vector representing the characteristics of a question.
- $(A, F) \rightarrow D$ refers to A and F jointly shape their historical response pattern D . Students with higher ability tend to provide correct answers more rapidly during the initial learning phase, while greater proficiency enables more consistent correct performance after mastery, collectively determining the observed distribution of interactions.
- $(A, F, D) \rightarrow S$ refers to modeling the process in conventional KT approaches, where the student representation S is learned by agglomerating the entire interaction history D , thereby entangling the distinct influences of intrinsic ability A , acquired proficiency F , and other student-specific traits.
- $(S, Q) \rightarrow Y$ denotes that the predicted response probability Y is determined by the joint effect of the student's features S and the question characteristics Q .

Crucially, ability and proficiency exert both direct and indirect effects on the student features:

- **Effects**: Ability and proficiency can directly impact the student features, independent of the current representation S (e.g., through factors like test-taking confidence or quick recall).
- **Effects (Mediated by S)**: More importantly, ability and proficiency shape the student representation S over time. This evolving representation S , in turn, directly influences the prediction Y . Therefore, the total effect of A and F on Y is composed of both their direct effects and their indirect effects mediated through S , accounting for their dual contributions to the outcome.

In summary, a student's performance is co-determined by their learning ability and practice-induced proficiency, which exhibit both complementarity (e.g., proficiency can compensate for lower ability) and transferability across related knowledge concepts or problem formats. However, conventional knowledge tracing models conflate these two distinct causal factors into a single, entangled representation of student ability. This conflation introduces confounding bias that hinders accurate prediction. By explicitly disentangling and separately modeling these two factors, we enable the model to isolate their distinct causal effects, thereby enhancing predictive accuracy.

Next, we will elucidate why distinguishing between learning ability and proficiency can lead to superior predictive performance from the perspective of probability modeling.

$$P(Y|S=s, Q=q) \quad (15a)$$

$$= \sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} \frac{P(Y, s, q, d, a, f)}{P(s, q)} \quad (15b)$$

$$= \sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} \frac{P(a)P(f)P(d|a, f)P(s|a, f, d)P(q)P(Y|s, q)}{P(s)P(q)} \quad (15c)$$

$$= \sum_{a \in \mathcal{A}} \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} P(a, f|d)P(d)P(Y|S(a, f, d), q) \quad (15d)$$

$$= P(a_s, f_s)P(Y|S(a_s, f_s, d_s), q) \quad (15e)$$

Eq.(15b) and Eq.(15c) are derived by applying the law of total probability, the structure of the causal graph, and Bayes' rule. Given that S is a deterministic function of A , F , and D , we obtain Eq.(15d). By focusing our analysis on the subspace d_s conditioned on specific ability a_s and proficiency f_s values, we arrive at the final estimable form in Eq.(15e), where $\mathcal{D}$, $\mathcal{A}$, $\mathcal{F}$ denote the sample spaces of D , A , F respectively.

The probabilistic derivation in Eq.(15e) demonstrates that the student feature S is confounded by the backdoor path $A, F \rightarrow D \rightarrow S$. This means the effects of intrinsic ability (A) and acquired proficiency (F) are entangled through the observed historical interaction pattern D . This dual-source confounding creates ambiguity in predictions: the model cannot determine if a high prediction stems from the student's ability, proficiency, or a mixture of both. Consequently, this entanglement introduces spurious correlations and adversely impacts prediction accuracy.

TABLE I: Statistics of 6 datasets.

Dataset	#students	#questions	#concepts	#interactions
Assist09	3,644	17,727	123	281,890
Assist17	1,079	3,162	102	942,816
Ed-Net	5,000	12,117	189	676,276
Spanish	182	409	221	578,726
Slepemapy	5,000	2,723	1,391	625,523
Algebra05	572	173,650	112	609,971

To address this, explicitly disentangling ability and proficiency allows the model to isolate their distinct causal effects. This separation breaks the spurious correlation, enabling a more accurate estimation of each factor’s true contribution to boost the performance.

V. EXPERIMENTS

A. Baselines

In this paper, we adopt the following 6 representative methods for KT as the baselines:

- **DKT**[4]: DKT is the first model that employs RNNs to establish a foundational deep learning baseline.
- **DKVMN**[10]: DKVMN introduces a memory-augmented neural network that uses a static key matrix and a dynamic value matrix to explicitly track and update students’ evolving knowledge states.
- **SAKT**[6]: SAKT employs a self-attention mechanism within a Transformer architecture to dynamically weigh past learning interactions.
- **SparseKT**[32]: SparseKT addresses the data sparsity challenge in KT by explicitly leveraging the sparsity of student interaction sequences.
- **SimpleKT**[22]: SimpleKT advocates for architectural simplicity and efficiency in KT by revisiting and simplifying the foundational DKT framework.
- **DisKT**[25]: DisKT pioneers the integration of causal inference into knowledge tracing to deconfound spurious correlations in learning data.

B. Datasets

Our experiments are conducted on 6 widely-used real-world datasets: Algebra05, Assistments09, Assistments17, Ed-Net, Slepemapy, and Spanish. The statistical information of the datasets after pre-processing is shown in Table I.

- **Algebra05**[36]: The Algebra05 dataset originates from the KDD Cup 2010 EDM Challenge and contains detailed step-level student responses to algebra questions.
- **ASSISTments09**[37]: The ASSISTments09 dataset is a foundational benchmark collected from the ASSISTment intelligent tutoring system during the 2009-2010 school year. It primarily consists of mathematics exercises and has been extensively utilized to evaluate the performance of knowledge tracing models.

- **ASSISTments17**[37]: The ASSISTments17 dataset has diverse problem types and inherent noise present a significant challenge for modeling complex, long-sequence learning.
- **Ed-Net**[38]: The Ed-Net dataset from the Santa tutoring platform, is the largest public educational interaction dataset, containing fine-grained logs of millions of student problem-solving attempts.
- **Slepemapy**[39]: The Slepemapy dataset from an adaptive geography practice platform, focuses on the factual recall of country locations, capitals, and flags.
- **Spanish**[40]: The Spanish dataset records middle-school students’ practice on vocabulary and grammar over a 15-week semester.

C. Implementation Details

We employ a comprehensive experimental setup to ensure robust evaluation. All models are evaluated using 5-fold cross-validation, where the data in each fold is decomposed into 80% for training, 10% for validation, and 10% for testing. The Adam optimizer is used for training all models. To mitigate overfitting, an early stopping strategy is applied, monitoring the validation performance with a patience of 10 epochs. To guarantee a fair comparison, we conduct a hyperparameter search for each model on every dataset. Key parameters, including the learning rate, dropout[41] rate, embedding dimension, and the hidden dimension, are tuned within a limited, predefined search space to identify the optimal configuration for each model-dataset pair based on validation set performance. All experiments are conducted on a Linux server using 2 NVIDIA GeForce RTX 3090 GPUs and all models are implemented in PyTorch. Our code is available at <https://github.com/ThaliaBee/PAKT>.

D. Performance Comparison

Following previous studies [25, 32], we adopt two metrics as our evaluation metrics: Area Under the ROC Curve (AUC) and Accuracy (ACC). The overall performances of all models, are summarized in Table II. Several observations can be made from the result as below.

As shown in Table II, PAKT achieves the best performance across all six datasets on both AUC and ACC metrics. The performance advantage spans diverse educational contexts, including mathematics (Algebra05, ASSISTments09, ASSISTments17), language learning (Spanish, Ed-Net), geography (Slepemapy). These improvements demonstrate the effectiveness of stage-based modeling principle for the knowledge tracing task. It is worth noting that DisKT [25], which also decomposes the input sequence, generally achieves competitive results compared to conventional KT models. However, this improvement is not stable: on certain datasets, DisKT underperforms relative to methods that do not employ sequence decomposing (e.g., SimpleKT and SparseKT). This phenomenon stems from the fact that a simple correctness-based split only provides a coarse separation of the interaction data, without capturing the temporal characteristics in learning process.

TABLE II: Performance comparison on all datasets. The reported values represent the mean from 5-fold cross-validation. The best performance is highlighted in bold, and the second-best is indicated with underlining. The relative improvement percentages achieved by PAKT are shown in the bottom row.

Dataset	Metric	DKT	DKVMN	SAKT	SparseKT	SimpleKT	DisKT	PAKT	%Improv.
Algebra05	AUC	0.7751	0.7824	0.7513	0.7882	0.7921	<u>0.7969</u>	0.7987	0.22%
	ACC	0.7961	0.7960	0.7834	0.7955	0.7953	<u>0.7995</u>	0.8017	0.27%
ASSISTments09	AUC	0.7590	0.7580	0.7305	0.7679	0.7730	<u>0.7775</u>	0.7812	0.49%
	ACC	0.7235	<u>0.7254</u>	0.7054	0.7210	0.7214	0.7220	0.7341	1.19%
ASSISTments17	AUC	0.6844	0.6781	0.6416	0.7126	0.7125	<u>0.7293</u>	0.7369	1.04%
	ACC	0.6534	0.6533	0.6324	0.6747	0.6738	<u>0.6830</u>	0.6872	0.62%
Ed-Net	AUC	0.6574	0.6611	0.6470	0.7033	0.7062	<u>0.7140</u>	0.7199	0.84%
	ACC	0.6402	0.6410	0.6301	0.6692	0.6661	<u>0.6714</u>	0.6754	0.60%
Slepemapy	AUC	0.7129	0.7130	0.6716	0.7261	<u>0.7306</u>	0.7299	0.7377	0.98%
	ACC	0.7883	0.7862	0.7802	<u>0.7916</u>	0.7910	0.7906	0.7952	0.46%
Spanish	AUC	0.8298	0.8250	0.8047	0.8366	0.8430	<u>0.8453</u>	0.8566	1.33%
	ACC	0.7774	0.7800	0.7609	0.7826	0.7893	<u>0.7904</u>	0.7998	1.19%

In contrast, PAKT introduces a fine-grained decomposing strategy. It explicitly models the temporal transition from ability phase to proficiency phase for each knowledge concept. Furthermore, the multi-branch backbone and the type-aware readout module fully leverage the complementary information from both phases. As a result, PAKT delivers stable and universal improvements over all baselines across every dataset. The AUC gains over the strongest baseline range from 0.22% to 1.33%.

The different magnitudes of improvement can be explained by different characteristics of datasets. On the one hand, PAKT achieves more stable and significant gains on datasets with more interactions and a moderate number of knowledge concepts. PAKT achieve +1.33% AUC on Spanish, +1.04% AUC on ASSISTments17, respectively. Spanish is characterized by moderate conceptual density and high conceptual interaction coverage, with students interacting with 213.8 out of 221 concepts on average. ASSISTments17 has the largest amount of interactions. In such settings, the phase shift from ability to proficiency is more clearly manifested in the data, allowing our stage-based decomposition to fully exploit its modeling capacity.

On the other hand, the smallest improvement is observed on Algebra05 (+0.22% AUC). We attribute this limited gain to the low concept coverage within each student’s interaction. A statistical analysis reveals that each Algebra05 student interacts with only 41.3 of its 112 concepts on average on the whole dataset. When very few concepts enter the proficiency phase, the proficiency branch receives sparse training signal. More than that, Algebra05 contains 173,650 questions, nearly ten times of ASSISTments09. This results in extremely sparse training signals per question, leaving the parameters related to question difficulty $\mathbf{d}_{q_t}$ in Eq.(3) severely undertrained and thus impairing its representational capacity.

E. Sensitivity Analysis of the Threshold Parameter k

Since the threshold k is a core hyper-parameter of our decomposing mechanism, we conducted experiments on the

value of k . k determines the dividing point between the ability and proficiency phases. Specifically, a larger k delays the transition point. We conduct a sensitivity analysis to understand the impact of k and to identify its optimal operating range. In our experimental setup, sequences are padded or truncated to a fixed length of 200, consistent with the main experiments. The results on all six datasets are visualized in Figure 5. Our findings are below:

First, across all datasets, performance peaks within a small range ($k \leq 5$). Setting k too high forces most interactions to remain in the ability phase, causing proficiency phase signal sparser.

Second, the optimal k varies across datasets. On Spanish and Algebra05, performance is highest at a small k and generally declines thereafter. Spanish ($k = 1$) exhibits a strictly monotonic decrease. In this domain with 221 concepts, slightly more than most datasets mentioned above, a student’s first correct attempt is already a strong signal of mastery. Raising k further only dilutes the sparse proficiency interactions with noise. Algebra05 exhibits a similar downward trend, although with a slight peak at $k = 2$. Due to its significantly fewer proficiency-active concepts, increasing k more severely reduces the number of interactions in the proficiency phase.

On ASSISTments09, ASSISTments17, and Ed-Net, AUC follows a clear rise-then-fall pattern. Very low k ($k = 1, 2$) leads to premature decomposing, while $k = 3$ provides sufficient evidence of stable mastery without over-delaying the transition.

Slepemapy nominally peaks at $k = 4$, but the AUC variation across all k values is the smallest among all datasets (range < 0.002), indicating that the model is largely insensitive to the threshold choice. The platform’s large concept pool, that is 1,391 concepts, severely dilutes the per-concept learning signal. Compounding this, its high average correct rate (80.9%) indicates that most presented concepts are already within a student’s competency range, which leaves little behavioral contrast between learning phases regardless of the threshold k .

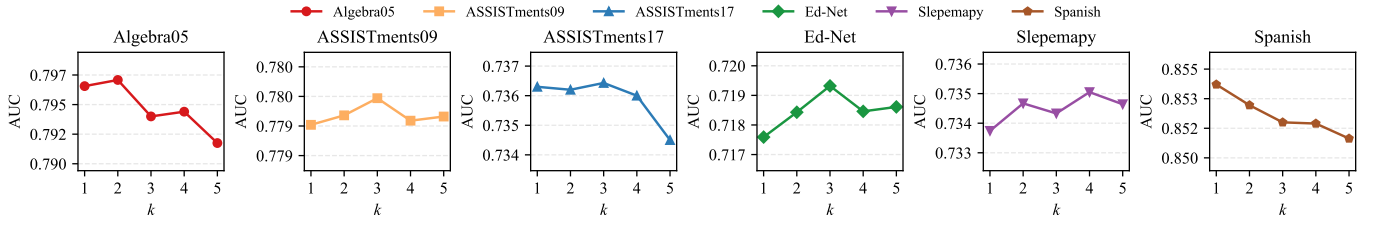
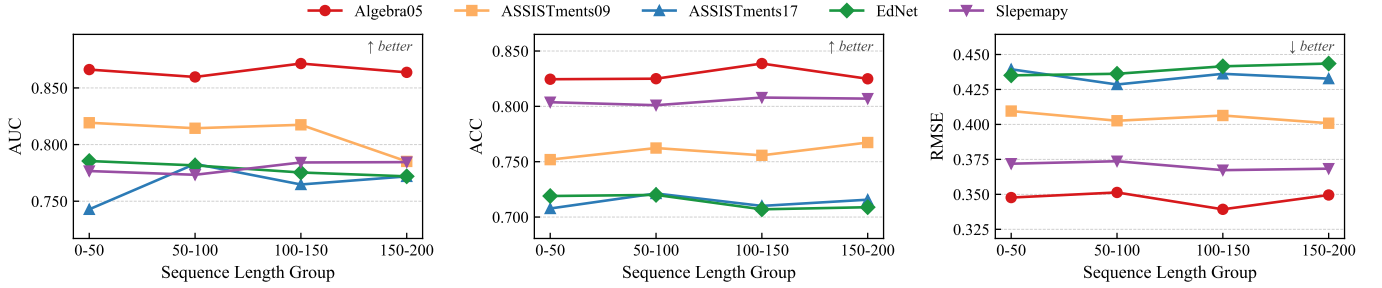
Fig. 5: Effect of parameter k on model performance .

Fig. 6: Performance of PAKT Across Different Sequence Length Groups.

F. Performance Across Different Sequence Length Groups

The effectiveness of PAKT is closely tied to sequence length, since it directly shapes how fully the ability-to-proficiency transition is observable within the modeled window.

The Spanish dataset is excluded, as all student sequences exceed the maximum input length and are truncated, placing every sample in the 150–200 group. Across nearly all datasets, the variation in AUC across length groups remains within approximately 0.03, indicating that the model maintains robust predictive performance. Within this overall stability, sequences in the 50–100 and 100–150 groups tend to yield the best performance, as they are long enough to reveal meaningful stage-based progression while remaining untruncated.

Sequences in the 0–50 group lack sufficient interactions to capture the complete ability-to-proficiency developmental trajectory. For sequences in the 150–200 group, the model truncates from the front to fit the maximum input length, discarding the student’s earliest learning interactions. This means the model loses access to early concept-acquisition behaviors, impairing its ability to characterize students’ initial ability-phase interactions. These findings confirm that a longer raw interaction history does not translate to better performance.

G. Ablation Study

To systematically investigate the contribution of each component in PAKT, we construct seven variants by selectively removing one or two branches from the full model, as shown in Fig 7.

As for the individual component contributions, generally speaking, removing the complete branch causes the largest average degradation among all single-branch ablations, from -0.45% to -1.38%, confirming it as the most indispensable component. The holistic interaction sequence preserves the

fundamental information in students’ interaction with the most abundant details. The phase-based branches cannot fully recover these details by reconstruction. Then, removing the ability branch or the proficiency branch also degrade the model performance on all datasets, which indicates the contributions of these branches for the KT task.

Moreover, as for the two-branch ablations, we can have the similar observations. For instance, retaining only the Complete branch (w/o ap) can achieve competitive performance on all datasets. Furthermore, it can be observed that retaining the Ability branch (w/o pc) yields better overall performance than retaining the Proficiency branch (w/o ac). This indicates that the ability branch is more important for characterizing students’ knowledge states.

Overall, these results validate the three-branch design: the Complete branch provides the foundational signal, while the Ability and Proficiency branches deliver complementary phase-aware refinements that neither can substitute for the other.

VI. CONCLUSION

In this paper, we propose PAKT, a KT model that incorporates two-phase learning strategy into the KT task to model the dynamics of knowledge acquisition and proficiency consolidation. PAKT decomposes each student’s interaction sequence into an ability phase and a proficiency phase, reflecting the behavioral semantics of distinct learning stages. A multi-branch Transformer backbone with a two-stage parameter-sharing mechanism then learns separate knowledge state representations from the ability, proficiency, and holistic branches. Student representations from different aspects are fused via a readout module for prediction. PAKT achieves consistent improvements in both AUC and ACC over all baselines across six public benchmarks.

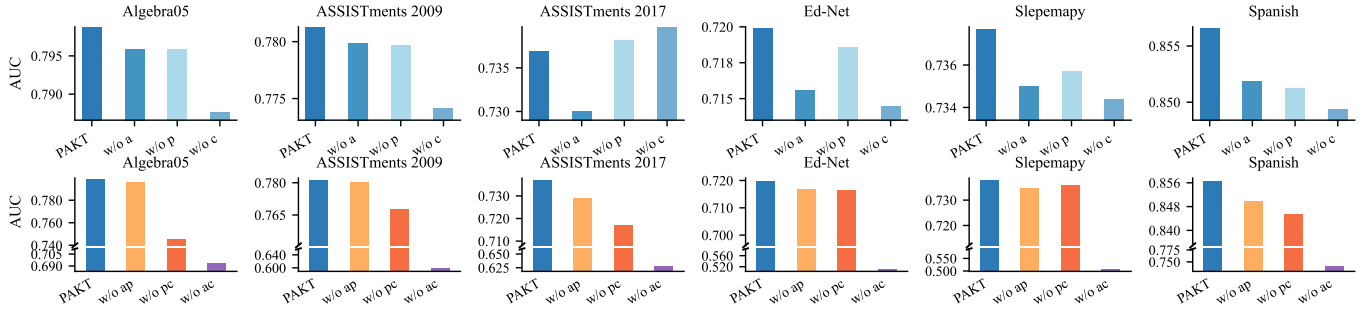


Fig. 7: Ablation study on 6 datasets.

However, PAKT has several limitations. First, when a student practices too few interactions with a knowledge concept, the cumulative correct response count does not reach the threshold k , leaving little or no proficiency-phase signal for that concept and thus weakening the proficiency branch’s effectiveness. Second, the optimal value of k must be manually tuned and varies across datasets. The observed AUC gains underscore the value of phase decomposition and motivate future investigation into automatic decomposition strategies as well as more robust handling of short interaction sequences.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62507035), the Wuhan University Undergraduate Education Quality Construction Comprehensive Reform Project (Future Curriculum), and the Teaching Reform Research Project of Wuhan University Engineering Management Project Center.

REFERENCES

- [1] Q. Tang, “A case study on the application of artificial intelligence in education industry,” in *Proceedings of the 2023 3rd International Conference on Modern Educational Technology and Social Sciences (ICMETSS 2023)*. Atlantis Press, 2023, pp. 99–109.
- [2] A. T. Corbett and J. R. Anderson, “Knowledge tracing: Modeling the acquisition of procedural knowledge,” *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.
- [3] F. M. Lord, *Applications of Item Response Theory to Practical Testing Problems*, 1st ed. New York: Routledge, 1980.
- [4] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, “Deep knowledge tracing,” in *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 505–513.
- [5] C. Yeung, “Deep-irt: Make deep learning based knowledge tracing explainable using item response theory,” in *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, Eds. International Educational Data Mining Society (IEDMS), 2019.
- [6] S. Pandey and G. Karypis, “A self attentive model for knowledge tracing,” in *Proceedings of the 12th International Conference on Educational Data Mining, EDM 2019, Montréal, Canada, July 2-5, 2019*, M. C. Desmarais, C. F. Lynch, A. Merceron, and R. Nkambou, Eds. International Educational Data Mining Society (IEDMS), 2019.
- [7] L. Fu, H. Guan, K. Du, J. Lin, W. Xia, W. Zhang, R. Tang, Y. Wang, and Y. Yu, “SINKT: A structure-aware inductive knowledge tracing model with large language model,” in *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, CIKM 2024, Boise, ID, USA, October 21-25, 2024*, E. Serra and F. Spezzano, Eds. ACM, 2024, pp. 632–642.
- [8] Y. Choi, Y. Lee, J. Cho, J. Baek, B. Kim, Y. Cha, D. Shin, C. Bae, and J. Heo, “Towards an appropriate query, key, and value computation for knowledge tracing,” in *L@S’20: Seventh ACM Conference on Learning @ Scale, Virtual Event, USA, August 12-14, 2020*, D. A. Joyner, R. F. Kizilcec, and S. Singer, Eds. ACM, 2020, pp. 341–344.
- [9] X. Lu, B. Long, X. Chen, Y. Han, and X. Sun, “I/O analysis is all you need: An I/O analysis for long-sequence attention,” in *Proceedings of the 31st ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2, ASPLOS 2026, Pittsburgh, PA, USA, March 22-26, 2026*, B. C. Lee, H. Xu, M. Silberstein, and B. Li, Eds. ACM, 2026, pp. 962–977.
- [10] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, “Dynamic key-value memory networks for knowledge tracing,” in *Proceedings of the 26th international conference on World Wide Web*, 2017, pp. 765–774.
- [11] H. Nakagawa, Y. Iwasawa, and Y. Matsuo, “Graph-based knowledge tracing: modeling student proficiency using graph neural network,” in *IEEE/WIC/ACM international conference on web intelligence*, 2019, pp. 156–163.
- [12] J. Cui, H. Qian, B. Jiang, and W. Zhang, “Leveraging pedagogical theories to understand student learning pro-

- cess with graph-based reasonable knowledge tracing,” in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, R. Baeza-Yates and F. Bonchi, Eds. ACM, 2024, pp. 502–513.
- [13] Q. Liu, Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, and G. Hu, “EKT: exercise-aware knowledge tracing for student performance prediction,” *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 1, pp. 100–115, 2021.
- [14] C. Huang, Q. Huang, X. Huang, H. Wang, M. Li, K. Lin, and Y. Chang, “XKT: toward explainable knowledge tracing model with cognitive learning theories for questions of multiple knowledge concepts,” *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 7308–7325, 2024.
- [15] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [16] J. Li, S. Mao, Y. Qin, F. Wang, and Y. Jiang, “Hyperbolic hypergraph transformer with knowledge state disentanglement for knowledge tracing,” *IEEE Trans. Knowl. Data Eng.*, vol. 37, no. 8, pp. 4677–4690, 2025.
- [17] J. Zhu, X. Ma, and C. Huang, “Stable knowledge tracing using causal inference,” *IEEE Trans. Learn. Technol.*, vol. 17, pp. 124–134, 2024.
- [18] S. Zu, L. Li, S. Cai, and J. Shen, “Cikt: Causality inspired knowledge tracing,” in *International Conference on Database Systems for Advanced Applications*. Springer, 2024, pp. 485–495.
- [19] J. Sun, F. Yu, Q. Wan, Q. Li, S. Liu, and X. Shen, “Interpretable knowledge tracing with multiscale state representation,” in *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, T. Chua, C. Ngo, R. Kumar, H. W. Lauw, and R. K. Lee, Eds. ACM, 2024, pp. 3265–3276.
- [20] J. Cui, M. Yu, B. Jiang, A. Zhou, J. Wang, and W. Zhang, “Interpretable knowledge tracing via response influence-based counterfactual reasoning,” in *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE, 2024, pp. 1103–1116.
- [21] C. Huang, H. Wei, Q. Huang, F. Jiang, Z. Han, and X. Huang, “Learning consistent representations with temporal and causal enhancement for knowledge tracing,” *Expert Syst. Appl.*, vol. 245, p. 123128, 2024.
- [22] Z. Liu, Q. Liu, J. Chen, S. Huang, and W. Luo, “simplekt: A simple but tough-to-beat baseline for knowledge tracing,” in *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [23] G. Abdelrahman, Q. Wang, and B. P. Nunes, “Knowledge tracing: A survey,” *ACM Comput. Surv.*, vol. 55, no. 11, pp. 224:1–224:37, 2023.
- [24] S. Shen, Q. Liu, Z. Huang, Y. Zheng, M. Yin, M. Wang, and E. Chen, “A survey of knowledge tracing: Models, variants, and applications,” *IEEE Trans. Learn. Technol.*, vol. 17, pp. 1898–1919, 2024.
- [25] Z. Yiyun, L. Zheqi, Z. Shengyu, and C. Jingyuan, “Disentangled knowledge tracing for alleviating cognitive bias,” in *Proceedings of the ACM on Web Conference*, 2025, pp. 2633–2645.
- [26] K. Zhang and Y. Yao, “A three learning states bayesian knowledge tracing model,” *Knowl. Based Syst.*, vol. 148, pp. 189–201, 2018.
- [27] S. P. Reise and N. G. Waller, “Item response theory and clinical measurement,” *Annual Review of Clinical Psychology*, vol. 5, no. Volume 5, 2009, pp. 27–48, 2009.
- [28] A. Ghosh, N. Heffernan, and A. S. Lan, “Context-aware attentive knowledge tracing,” in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 2330–2339.
- [29] W. Lee, J. Chun, Y. Lee, K. Park, and S. Park, “Contrastive learning for knowledge tracing,” in *Proceedings of the ACM web conference 2022*, 2022, pp. 2330–2338.
- [30] Y. Yin, L. Dai, Z. Huang, S. Shen, F. Wang, Q. Liu, E. Chen, and X. Li, “Tracing knowledge instead of patterns: Stable knowledge tracing with diagnostic transformer,” in *Proceedings of the ACM web conference 2023*, 2023, pp. 855–864.
- [31] C. Cui, H. Ma, C. Zhang, C. Zhang, Y. Yao, M. Chen, and Y. Ma, “Do we fully understand students’ knowledge states? identifying and mitigating answer bias in knowledge tracing,” *CoRR*, vol. abs/2308.07779, 2023.
- [32] S. Huang, Z. Liu, X. Zhao, W. Luo, and J. Weng, “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, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023*, H. Chen, W. E. Duh, H. Huang, M. P. Kato, J. Mothe, and B. Poblete, Eds. ACM, 2023, pp. 2441–2445.
- [33] P. M. Fitts and M. I. Posner, *Human Performance*. Belmont, CA, USA: Brooks/Cole Publishing Company, 1967, includes bibliographical references (pages 151–158).
- [34] Y. Zheng, C. Gao, X. Li, X. He, Y. Li, and D. Jin, “Disentangling user interest and conformity for recommendation with causal embedding,” in *Proceedings of the web conference 2021*, 2021, pp. 2980–2991.
- [35] W. Wang, F. Feng, X. He, X. Wang, and T.-S. Chua, “Deconfounded recommendation for alleviating bias amplification,” in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 1717–1725.
- [36] J. Stamper, A. Niculescu-Mizil, S. Ritter, G. Gordon, and K. R. Koedinger, “Algebra I 2005-2006 and Bridge to Algebra 2006-2007. Development data sets from KDD Cup 2010 Educational Data Mining Challenge,” 2010.
- [37] M. Feng, N. T. Heffernan, and K. R. Koedinger, “Addressing the assessment challenge with an online system that tutors as it assesses,” *User Model. User Adapt. Interact.*, vol. 19, no. 3, pp. 243–266, 2009.
- [38] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo, “Ednet: A large-scale hierarchical dataset in education,” in *Artificial Intelligence in Education - 21st International Conference, AIED 2020, Ifrane, Morocco, July 6-10, 2020, Proceedings, Part II*,

- ser. Lecture Notes in Computer Science, I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, and E. Millán, Eds., vol. 12164. Springer, 2020, pp. 69–73.
- [39] J. Papousek, R. Pelánek, and V. Stanislav, “Adaptive geography practice data set,” *J. Learn. Anal.*, vol. 3, no. 2, pp. 317–321, 2016.
- [40] R. V. Lindsey, M. Khajah, and M. C. Mozer, “Automatic discovery of cognitive skills to improve the prediction of student learning,” in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 1386–1394.
- [41] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.