Tracing Mathematical Proficiency Through Problem-Solving Processes

Jungyang Park[1,2*] **Suho Kang**[3*] **Jaewoo Park**[1]
Jaehong Kim[2] **Jaewoo Shin**[2] **Seonjoon Park**[2] **Youngjae Yu**[3]

[1]Yonsei University [2]Mathpresso [3]Seoul National University
wjddid000624@yonsei.ac.kr youngjaeyu@snu.ac.kr

Abstract

Knowledge Tracing (KT) aims to model student's knowledge state and predict future performance to enable personalized learning in Intelligent Tutoring Systems. However, traditional KT methods face fundamental limitations in explainability, as they rely solely on the response correctness, neglecting the rich information embedded in students' problem-solving processes. To address this gap, we propose Knowledge Tracing Leveraging Problem-Solving Process (KT-PSP), which incorporates students' problem-solving processes to capture the multidimensional aspects of mathematical proficiency. We also introduce KT-PSP-25, a new dataset specifically designed for KT-PSP. Building on this, we present StatusKT, a KT framework that employs a teacher-student-teacher three-stage LLM pipeline to extract students' Mathematical Proficiency (MP) as intermediate representation. In this pipeline, the teacher LLM first extracts problem-specific proficiency indicators, then a student LLM generates responses based on the student's solution process, and a teacher LLM evaluates these responses to determine mastery of each indicator. The experimental results on KT-PSP-25 demonstrate that StatusKT improves the prediction performance of existing KT methods. Moreover, StatusKT provides interpretable explanations for its predictions by explicitly modeling students' mathematical proficiency. Code is available here.

1 Introduction

Knowledge Tracing (KT) is a technique that models a learner's evolving knowledge state over time (Corbett and Anderson, 1994; Liu et al., 2025). Since the true knowledge state cannot be observed directly, KT instead predicts future correctness from historical interaction data, as illustrated in Figure 1 (a). Early KT approaches, such as Bayesian Knowledge Tracing (BKT) (Corbett and Anderson, 1994) and Item Response Theory (IRT) (Green Jr, 1951), introduced interpretable probabilistic frameworks but struggled to represent complex learning behavior due to simplifying assumptions (e. g. , skill independence and coarse latent dynamics). With the advent of neural networks, deep learning-based KT (DLKT) models have demonstrated notable improvements by employing architectures such as recurrent neural networks (Piech et al., 2015; Nagatani et al., 2019) and attention mechanisms (Pandey and Karypis, 2019; Ghosh et al., 2020). Beyond architectural innovations, subsequent studies have further enriched KT models by integrating diverse contextual signals, such as item difficulty (Yeung, 2019; Chen et al., 2023), and knowledge concept structure (Su et al., 2021), thereby enhancing their performance and interpretability.

Although KT has advanced in various ways, most approaches rely heavily on outcome-centric supervision (e. g. , Correctness) and limited contextual metadata (e. g. , Knowledge Concepts (KCs), problem-response time, and difficulty), which often fail to reflect students' underlying understanding (Wang and Heffernan, 2012; Tschisgale et al., 2025; Yeung, 2019; Liu et al., 2024). In contrast, as shown in Figure 1 (b), real-world teachers evaluate students' understanding through the problem-solving process (Chiu et al., 2022; Tschisgale et al., 2025; Kleinman et al., 2022). Nevertheless, despite the importance of the solution process, current KT approaches and datasets remain underexplored for incorporating it (Feng et al., 2009; Liu et al., 2023b; Kim et al., 2025).

To address this limitation, we propose KNOWLEDGE TRACING WITH PROBLEM-SOLVING PROCESS (KT-PSP), a new formulation of the KT task that explicitly incorporates students' problem-solving processes (PSP) into the interaction sequence. To facilitate research on KT-PSP, we in-

[*] Equal Contribution.

Findings of the Association for Computational Linguistics: ACL 2026, pages 19251–19269
July 2-7, 2026 ©2026 Association for Computational Linguistics

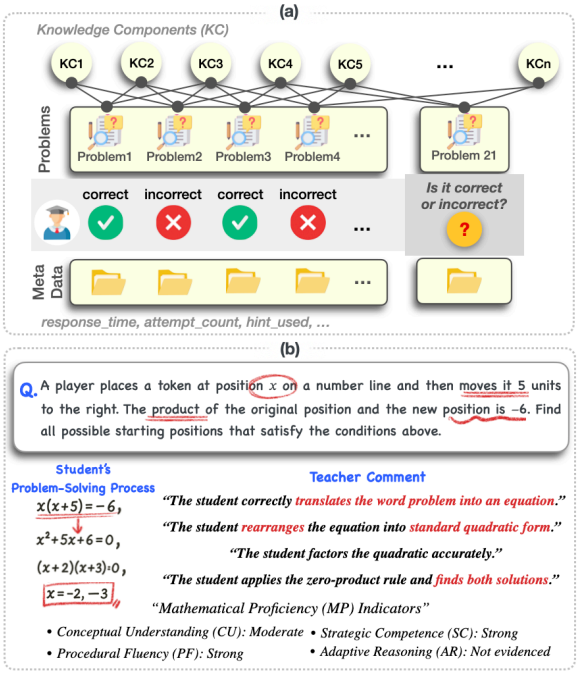

Figure 1: (a) Conventional KT utilizes limited metadata, lacking interpretability. (b) By incorporating problem context, students' reasoning, and teacher comments, StatusKT captures granular proficiency signals aligned with teachers' assessment

troduce KT-PSP-25, a novel dataset that contains real-world PSP for each student-problem interaction.

Building upon this dataset, we introduce StatusKT, a process-aware KT framework that uses a three-stage LLM pipeline to extract students' Mathematical Proficiency (MP) from their PSP and use it as an intermediate signal for modeling student knowledge. Inspired by El-Shara et al. (2025), StatusKT adopts a teacher-student-teacher structure: a teacher LLM derives problem-specific MP indicators, a student LLM answers them based on the students' written PSP, and a teacher LLM evaluates these answers to produce MP signals. These MP signals are then integrated into the KT backbone as auxiliary inputs to improve predictive accuracy and interpretability. Experiments on KT-PSP-25 demonstrate consistent improvements over strong DLKT baselines and further alleviate the cold-start problem by providing informative MP signals even in early interactions. In addition to a higher predictive accuracy, StatusKT provides interpretable proficiency signals that offer fine-grained insights into students' learning progress.

The major contributions of this paper are as follows:

- We introduce KT-PSP, a new KT task formulation that incorporates students' PSP into the interaction sequence, enabling process-aware modeling of knowledge.
- We release KT-PSP-25, a new mathematical KT dataset containing real-world PSP for each student-problem interaction.
- We propose StatusKT, a novel KT framework that extracts MP signals from students' PSP through teacher-student-teacher LLM pipeline and integrates them as auxiliary representations for KT.
- Experiments on KT-PSP-25 show that StatusKT consistently improves prediction performance over DLKT baselines while providing interpretable MP-aligned signals.

2 Related Works

2.1 Knowledge Tracing

KT plays a crucial role in intelligent tutoring systems by estimating students' evolving knowledge states from their interaction histories. A common line of work infers concept mastery in an autoregressive manner using past question-response data (Piech et al., 2015; Yeung and Yeung, 2018; Nagatani et al., 2019; Guo et al., 2021; Zhou et al., 2025). Memory-augmented models further encode concepts using a static key memory for soft attention and a dynamic value memory to update students' knowledge states (Zhang et al., 2017; Abdelrahman and Wang, 2019). Graph-based approaches (Nakagawa et al., 2019; Yang et al., 2020) capture the structural relations between questions and KCs, propagating historical information via graph updates. Attention-based KT approaches (Pandey and Karypis, 2019; Choi et al., 2020a; Ghosh et al., 2020; Liu et al., 2023a; Li et al., 2024) improve prediction by selectively focusing on informative past interactions.

Despite these advances, KT models remain sensitive to noisy behavioral traces. Therefore, recent studies have separated stable cognitive patterns from anomalous outcomes, such as slips (Guo et al., 2025). However, key determinants remain underexplored, most notably students' PSP and the MP they reveal. Accordingly, we propose a method that leverages fine-grained signals from students' PSP to infer their MP (Findell et al., 2001; Sullivan, 2011), thereby enhancing knowledge tracing.

2.2 Knowledge Tracing Dataset

The ASSISTments datasets (Feng et al., 2009; Pardos et al., 2014) served as early large-scale datasets for KT models. Subsequent datasets, such as Junyi 2015 (Chang et al., 2015), extended this line of research by leveraging online learning logs. In parallel, researchers began to evaluate students' performance rather than correctness alone, as reflected in initiatives such as the KDD Cup 2010 (KDD 2010, 2010). However, most early datasets contained only simple question-answer sequences, offering little auxiliary information. Over time, KT datasets have expanded beyond mathematics to other domains, such as English, computer science, engineering, and early childhood education (Choi et al., 2020b; Abdelrahman et al., 2022; Kim et al., 2025), motivating richer multimodal and cross-domain learning datasets. Recent efforts by Liu et al. (2023b) incorporated question content, answer explanations, and hierarchical KC structures to support a more comprehensive evaluation.

However, students' PSP remains largely absent, particularly in mathematical problems, where it is difficult to capture at scale. To bridge this gap, we constructed the first dataset that explicitly recorded PSP.

3 Knowledge Tracing with Problem-Solving Process

Knowledge Tracing (KT) is a task in educational data mining that aims to model a student's knowledge state over time based on their historical learning interactions. The primary goal is to predict how a student will perform on a future question. Formally, given a student's interaction history sequence $\mathcal{S}_{conv}$:

$$\mathcal{S}_{conv} = \{(q_1, c_1, r_1), \ldots, (q_t, c_t, r_t)\}, \quad (1)$$

where q_t denotes the question at time t, c_t is the concept associated with q_t, and $r_t \in \{0, 1\}$ represents the response correctness (1 for correct, 0 for incorrect), the objective is to estimate $P(r_{t+1} = 1 \mid q_{t+1},\ c_{t+1},\ \mathcal{S}_{conv})$.

However, relying solely on response correctness (r_t) creates a simplified proxy for students' understanding, often failing to capture the nuances of their actual knowledge state. In real educational environments, students are evaluated not only on the final answer but also on their problem-solving processes (PSP), which provides richer insight into students' understanding (Ukobizaba et al., 2021; Chiu et al., 2022; Tschisgale et al., 2025; Kleinman et al., 2022).

To overcome the limitations of binary outcomes, recent studies have attempted to incorporate auxiliary information into the KT models. For instance, response time has been widely used to distinguish between guessing and slipping behaviors (Wang and Heffernan, 2012; Chen et al., 2022; Huang et al., 2024). Others have integrated question difficulty or textual content to better estimate the probability of correctness based on item characteristics (Yeung, 2019; Liu et al., 2024). Although these approaches enrich the context, they largely rely on metadata or static attributes, treating the actual PSP as a black box.

Therefore, we propose Knowledge Tracing with Problem-Solving Process (KT-PSP), which extends the traditional KT paradigm by incorporating students' PSP into the interaction history sequence $\mathcal{S}_{new}$:

$$\mathcal{S}_{new} = \{(q_1, c_1, r_1, p_1), \ldots, (q_t, c_t, r_t, p_t)\}, \quad (2)$$

where p_t denotes the detailed PSP for question q_t, including step-by-step reasoning, intermediate computations, or other process-level traces. The objective remains estimating $P(r_{t+1} = 1 \mid q_{t+1},\ c_{t+1},\ \mathcal{S}_{new})$, while now leveraging both historical performance and the underlying problem-solving strategies.

4 Dataset

4.1 License

KT-PSP-25 is released under the CC BY-NC 4.0 (Creative Commons Attribution-NonCommercial 4.0 International) license. This license restricts the dataset to noncommercial use while allowing modifications and sharing under similar terms. Under this license, others can non-commercially remix, adapt, and build upon the work if they credit the source and indicate if changes were made.

4.2 Data Construction

We curate 22,289 problem-solving sessions from an in-house tablet-based mathematics education platform. Each session corresponds to a single student working on one problem, with interactions recorded between November 2024 and July 2025. To construct the dataset, we applied the following preprocessing steps: (1) remove sessions with fewer than five handwritten lines and (2) discard

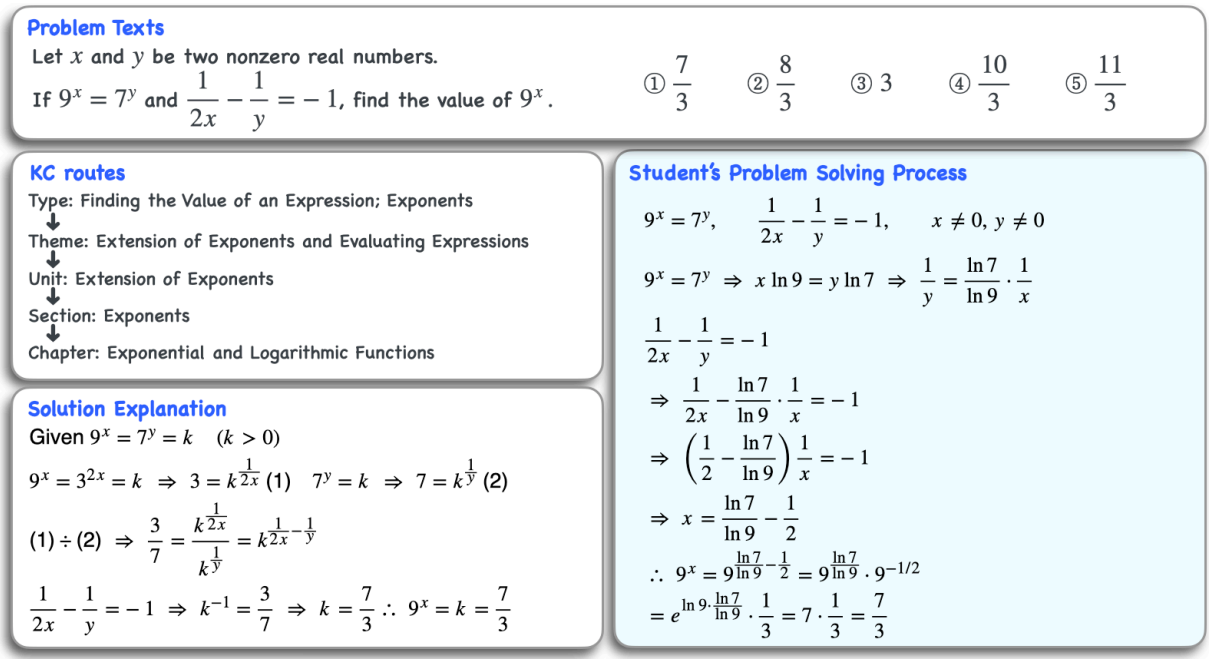

Figure 2: **An example question from KT-PSP-25 (problem text and KC translated into English).** For short-answer questions, no answer choices are provided.

Number of interactions	22,289
Number of students	1,343
Number of questions	2,696
Number of Knowledge Components	490
Average solution length (words)	73.54
Average PSP length (words)	24.76
Average correct ratio per student	0.72
Average session duration	60.36

Table 1: **Statistics of our dataset**

problems with missing textual content. The resulting dataset, illustrated in Figure 2, encompasses a wide range of information, including problem-level attributes (problem ID, associated knowledge concepts (KCs), problem text, solution explanation, ground-truth answer, question type, and difficulty) and student interaction attributes (selected answer, duration, PSP, and final correctness).

4.3 Data Privacy

Beyond these attributes, KT-PSP-25 also contains rich records of students' interaction sequences, including student IDs, question IDs, and handwritten PSP. Since such data may expose personal information, we implemented privacy-preserving measures. Student and question IDs were mapped to non-reversible digital identifiers to ensure anonymity. For handwritten PSP, we applied an OCR pipeline based on GPT-5 (OpenAI, 2025). After initial transcription, we further leveraged GPT to evaluate and refine the OCR outputs, thereby improving both privacy protection and data quality. Detailed prompts and discussion of OCR quality are provided in Appendix D

4.4 Data Analysis

There are 22,289 interactions, and 1,343 students answered 2,696 problems from 490 KCs. On average, the reference solution contains 73.54 words while the students' PSP contains 24.76 words, providing substantive reasoning traces rather than short answer-only logs.

Each record additionally stores both the start and completion times, enabling the computation of response durations and temporal analyses across learning sessions. Questions are assigned to five difficulty levels, with most items concentrated around medium difficulty (level 50), while very easy (10) and very hard (90) items appear infrequently. Summary statistics, including the process length, duration, and difficulty breakdown, are reported in Table 1.

5 Methodology

This section presents the details of the proposed StatusKT, as illustrated in Figure 3. StatusKT enhances knowledge tracing by explicitly modeling students' mathematical proficiency (MP) through

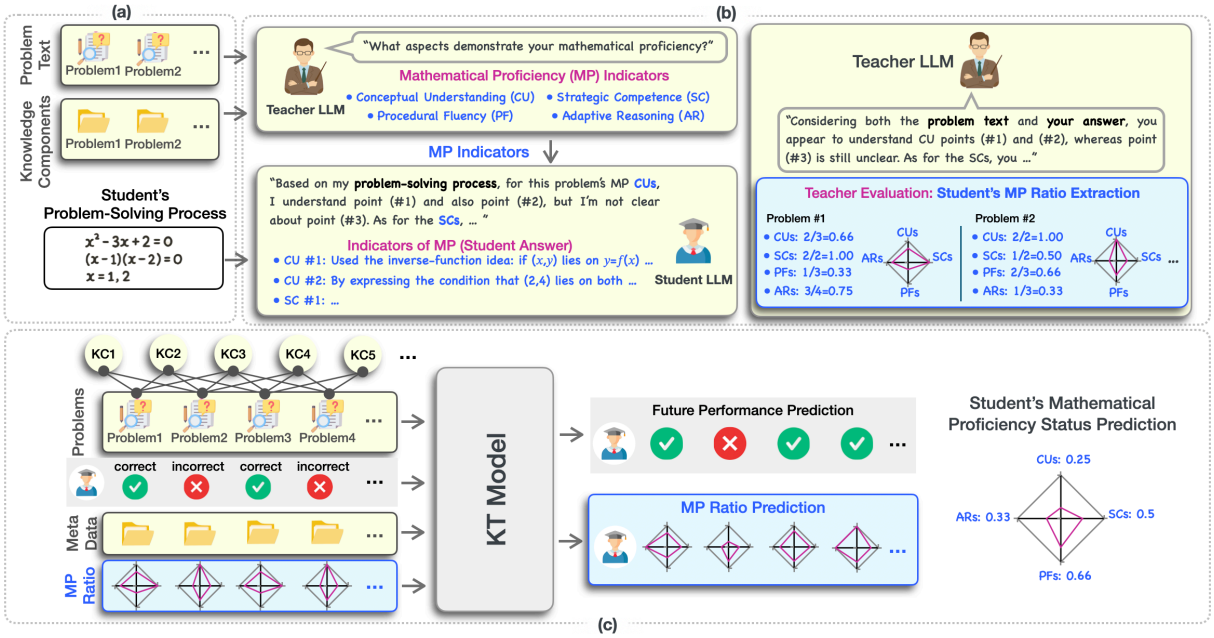

Figure 3: **StatusKT Framework Overview.** (b) A teacher LLM analyzes the text and knowledge components of each problem to construct MP indicators, whereas a student LLM identifies which it understands and can explain. Using these responses, the teacher LLM integrates the results to generate an MP ratio quantifying each student's Mathematical Proficiency(MP). (c) Using the MP ratio, the KT model predicts both students' future performance and problem-level MP ratios, offering a strong rationale for proficiency assessment.

interpretable signals derived from their problem-solving process (PSP). While conventional KT models rely on binary correctness, our approach incorporates structured proficiency evidence generated from the PSP. Although Findell et al. (2001) define MP as comprising five key strands–conceptual understanding (CU), strategic competence (SC), procedural fluency (PF), adaptive reasoning (AR), and productive disposition(PD)–we focus on four observable dimensions, excluding PD because it is difficult to assess reliably through textual problem-solving processes.

5.1 Deriving MP Indicators from Problem-Solving Processes

The input to our framework includes OCR-transcribed handwritten solution processes (Figure 3 (a)). These traces contain interpretable signals of students' reasoning capabilities. Our pipeline is inspired by the structured assessment protocol in El-Shara et al. (2025), in which students respond to a set of predefined MP indicators and mastery is determined based on whether those indicators are satisfied. Similarly, our framework adopts an indicator-based evaluation paradigm. However, because we cannot rely on fixed indicators or direct student interviews, we employ a three-stage Teacher-Student-Teacher LLM pipeline (Figure 3 (b)) to automatically extract MP from the raw problem-solving processes:

(i). Indicator Extraction(Teacher LLM). Given the problem q and its associated concept c, the first Teacher LLM generates a set of MP indicators:

$$\mathcal{I}_q = f_T(q, c) = \{I_1, \cdots, I_K\}, \tag{3}$$

where each I_k denotes individual indicators phrased as a question targeting a specific proficiency dimension. These indicators act as a rubric specifying the reasoning elements required for the problem, guiding the subsequent analysis.

(ii). Response Generation (Student LLM). Given the MP indicators $\mathcal{I}_q$ and a student's problem-solving process p, the Student LLM generates responses for each indicator:

$$\mathrm{R}_k = f_S(I_k, p). \tag{4}$$

This step converts implicit reasoning in the problem-solving process into explicit evidence corresponding to each proficiency dimension.

(iii). Proficiency Assessment (Teacher LLM). Finally, the second teacher LLM evaluates whether each response satisfies its corresponding indicator, producing a binary decision:

$$\mathrm{E}_k = f_T(I_k, R_k), \quad E_k \in \{0, 1\}. \tag{5}$$

Based on these evaluations, we compute the MP Ratio for each proficiency dimension d as the proportion of satisfied indicators:

$$\mathrm{MP}_d = \frac{1}{|\mathcal{I}_d|} \sum_{k \in \mathcal{I}_d} E_k, \tag{6}$$

where $\mathcal{I}_d \subseteq \mathcal{I}_q$ denotes the subset of indicators corresponding to dimension d.

In all three stages, we utilized GPT-5 (OpenAI, 2025) model. The specific prompts used in the pipeline are provided in Appendix E.2, and the human evaluation protocol and results are summarized in Appendix E.1.

5.2 Integrating MP Ratio for Interpretable Knowledge Tracing

The computed MP ratio serves as an interpretable intermediate representation that complements the traditional KT signals. As depicted in Figure 3 (c), the KT backbone utilizes these ratios to predict both (i) students' future performance on subsequent problems and (ii) problem-level MP ratios, enabling fine-grained proficiency tracking. This auxiliary MP prediction encourages the model to internalize proficiency-related patterns, enhancing both the predictive performance and interpretability beyond what correctness alone provides.

To train the model to effectively capture these multidimensional proficiency signals, we defined a composite loss function. The total loss is a weighted sum of the correctness prediction loss and proficiency regression loss:

$$L = \mathrm{BCE}(r_{\text{gt}}, r_{\text{pred}}) + \alpha \sum_{i \in P} \mathrm{MSE}(m_i^{\text{gt}}, m_i^{\text{pred}}), \tag{7}$$

where $P = \{\mathrm{CU}, \mathrm{SC}, \mathrm{PF}, \mathrm{AR}\}$, r denotes the response correctness, and m represents the MP ratio for dimension i. Hyperparameter α controls the trade-off between the correctness prediction and proficiency estimation. Through this dual prediction setup, StatusKT improves predictive accuracy while providing interpretable proficiency assessments that align with the established theories of mathematical learning.

6 Experiments

In this section, we conducted experiments on the KT-PSP-25 to evaluate our proposed StatusKT framework. Specifically, we aim to address the following research questions:

- **RQ1:** Does StatusKT outperform conventional DLKT baselines and text-embedding process baselines?
- **RQ2:** Does StatusKT improve robustness under cold-start conditions?
- **RQ3:** How do MP signals affect KT predictions–when do they help, when do they hurt, and why?

6.1 Experiment Setting

6.1.1 Baseline

We evaluate the performance of StatusKT with KT models, including DKT (Piech et al., 2015), DKT+ (Yeung and Yeung, 2018), DKVMN (Zhang et al., 2017), SKVMN (Abdelrahman and Wang, 2019), SAKT (Pandey and Karypis, 2019), SAINT (Choi et al., 2020a), AKT (Ghosh et al., 2020), SimpleKT (Liu et al., 2023a), StableKT (Li et al., 2024), RobustKT (Guo et al., 2025).

To isolate the impact of incorporating problem-solving process information without MP modeling, we introduce an additional baseline. In this setting, the encoded textual problem-solving process is concatenated with the original KT input features, allowing the model to access raw process signals without explicit MP reasoning.

6.1.2 Implementation Details

The dataset was split by students into 80% for training/validation and 20% for testing. To obtain reliable results given the limited dataset size, we performed 10-fold cross-validation on the training/validation data and selected the best model using early stopping with a patience of 10 epochs. All models were trained using the ADAM optimizer (Adam et al., 2014) with a batch size of 16.

For the text-embedding baseline, we utilized the `sentence-transformers/all-mpnet-base-v2` model to encode problem-solving processes and project the resulting representations into a 128-dimensional vector.

We follow standard KT training practices; full hyperparameter grids and implementation settings are listed in Appendix C.

6.1.3 Evaluation Metrics

We evaluated the model performance using two standard metrics in knowledge tracing: (1) Accuracy (ACC), which measures the proportion of correctly predicted student responses, and (2) Area under the ROC Curve (AUC), which captures the

Method	DKT	DKT+	DKVMN	SKVMN	SAKT	SAINT	AKT	simpleKT	stableKT	robustKT
AUC										
Baseline	0.6165	0.6192	0.6049	0.5866	0.5819	0.6201	0.6524	0.6591	0.6735	0.6373
Baseline+*PSP*	0.6135	0.6169	0.6001	0.6037	**0.6074**	0.6230	0.6457	0.6465	0.6635	**0.6443**
StatusKT	**0.6197**	**0.6200**	**0.6220**	**0.6040**	0.5854	**0.6401**	**0.6629**	**0.6639**	**0.6773**	0.6419
ACC										
Baseline	0.7219	0.7226	0.7292	0.7189	0.7243	0.7234	0.7396	0.7393	0.7345	0.7370
Baseline+*PSP*	0.7209	0.7233	0.7359	**0.7335**	**0.7385**	0.7326	0.7330	0.7288	0.7219	0.7288
StatusKT	**0.7235**	**0.7247**	**0.7410**	0.7257	0.7313	**0.7377**	**0.7435**	**0.7459**	**0.7421**	**0.7394**

Table 2: **Results of the main experiment.** We compare performance across diverse KT architectures under three settings: (i) original baselines, (ii) text-embedded *PSP* variants(which concatenate an encoded problem-solving process with the KT inputs without explicit MP reasoning), and (iii) the proposed StatusKT framework. Bold indicates the best result, and underlining indicates the second best.

	DKT	DKT+	DKVMN	SKVMN	SAKT	SAINT	AKT	simpleKT	stableKT	robustKT
Paired t-test (p)	0.0003	0.4894	2.47e-07	0.0004	0.0843	9.84e-06	2.27e-05	0.0410	0.0018	0.0489
Cohen's d	1.7888	0.2279	4.3331	1.7144	0.6135	2.7972	2.5221	0.7539	1.3873	0.7196

Table 3: Statistical significance of AUC improvements (Baseline vs StatusKT) across 10-fold cross-validation.

ranking quality of the predicted correctness probabilities. Following prior KT studies, the AUC was considered the primary metric because of its robustness to label imbalance across the student interactions.

6.2 Overall Performance (RQ1)

To address our first research question (RQ1) concerning the effectiveness of the StatusKT framework on KT-PSP-25 with students' problem-solving processes, we applied it to a wide range of existing KT architectures.

The experimental results are summarized in Table 2. Analysis of both AUC and ACC metrics reveals that merely appending raw text embeddings to the original models (Baseline+*PSP*) does not consistently enhance performance. In several architectures, the performance fluctuates around the baseline, suggesting that the inclusion of process text may introduce noise. We hypothesize that this instability stems from OCR artifacts and the mismatch between free-form textual traces and the discrete KT objective, motivating our structured MP distillation. In contrast, models augmented with StatusKT achieve consistent improvements over both the baseline and Baseline+*PSP* for most architectures. The enhancements are particularly notable in RNN-based models (e.g., DKT) and recent transformer-style approaches (e.g., SAINT, AKT), which exhibit higher AUC and ACC with our method. These findings suggest that our StatusKT framework distills the process into structured indicators that can be effectively integrated with the interaction history.

To verify that these improvements are not artifacts of fold-level fluctuations, we conducted paired t-tests across the 10-fold splits, comparing StatusKT with each corresponding baseline under the same data partitions. The results are summarized in Table 3. The results confirm that the gains are statistically significant ($p < 0.05$) for the majority of architectures, with moderate to large effect sizes (Cohen's d). Two models, DKT+ and SAKT, do not reach significance, which aligns with their relatively low baseline AUC where the MP signals offer limited additional leverage.

Overall, these findings address RQ1, indicating that unprocessed PSP alone is insufficient for improving KT performance, whereas our framework consistently delivers performance improvements through principled process modeling.

6.3 Performance in Cold-Start (RQ2)

Cold-start arises when models must predict performance for unseen students with only a few early interactions. With such limited evidence, estimating learner's latent knowledge states becomes unstable and often degrades predictive performance (Slater and Baker, 2018; Bhattacharjee and Wayllace, 2025). Prior work (Guo et al., 2024; Bai et al., 2025) introduces architectural priors or external semantics, yet reliably estimating a student's state from sparse correctness observations remains difficult. Our StatusKT framework in-

	DKT	DKT+	DKVMN	SKVMN	SAKT	SAINT	AKT	SimpleKT	StableKT	RobustKT
AUC										
Baseline	**0.6072**	**0.6093**	**0.5875**	0.5758	0.5764	0.6228	0.6217	0.6406	0.6516	0.6196
StatusKT	0.6031	0.6069	0.5843	**0.5826**	**0.5818**	**0.6266**	**0.6352**	**0.6442**	**0.6541**	**0.6302**
ACC										
Baseline	0.7064	0.7107	0.7043	0.7207	0.7113	0.7350	0.7189	0.7219	0.7308	0.7330
StatusKT	**0.7146**	**0.7134**	**0.7083**	**0.7253**	**0.7230**	**0.7471**	**0.7278**	**0.7261**	**0.7538**	**0.7402**

Table 4: **Cold-start performance ($t \leq 5$).** We evaluate each KT architecture using only the first five observed interactions per student. Models augmented with StatusKT generally achieve higher AUC and ACC, indicating that MP signals help estimate learners' knowledge states even under limited early evidence.

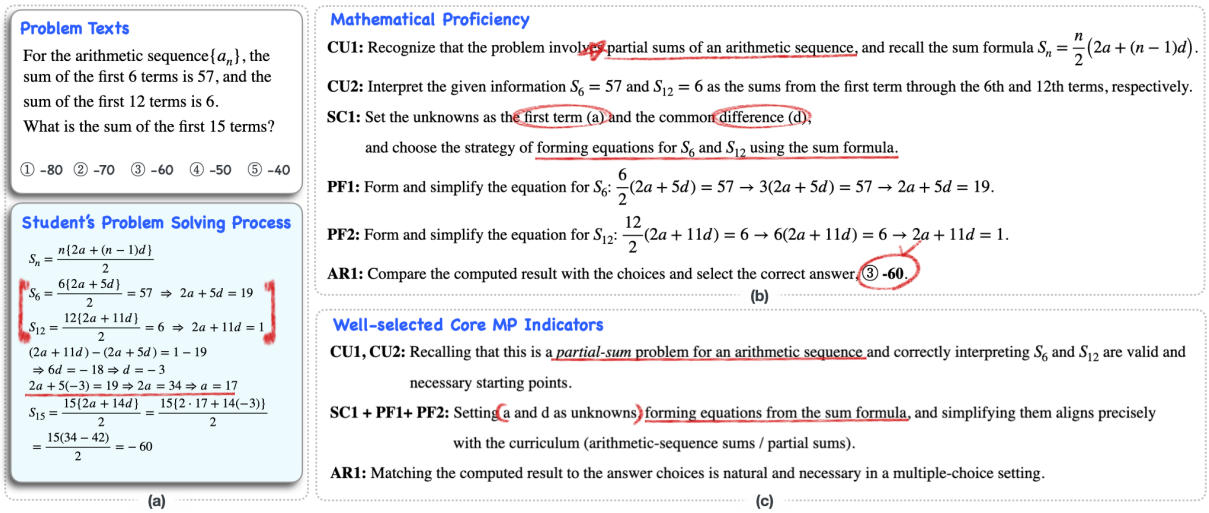

Figure 4: **Example MP annotation and assessment used for case inspection.** (a) Problem text and student's PSP. (b) Extracted MP indicators. (c) Rationale-based evaluation of MP scope.

stead leverages process-derived MP signals, which begin appearing even within the first few interactions and thus offer richer early information. To examine whether MP signals mitigate cold-start, we evaluate models under a strict student-level hold-out protocol and restrict each sequence to the first five interactions ($t \leq 5$). As reported in Table 4, incorporating StatusKT framework consistently improves ACC across all architectures and yields competitive or higher AUC in most cases. These gains indicate that MP signals provide informative priors that stabilize state estimation when early data are scarce, supporting our hypothesis that process modeling is particularly beneficial in low-evidence regimes.

6.4 Analysis of MP Effects on KT (RQ3)

To examine whether MPs provide a useful representation for KT, we conduct a case-based diagnostic analysis that contrasts predictions made with and without MP. Following the template in Figure 4, we inspect each case by aligning the problem text and the student's PSP with the corresponding MPs. We focus on three patterns: (i) *MP-help* instances where adding MP flips an incorrect prediction to correct, (ii) *confidence-gap analysis* within MP-help cases to understand why predicted probabilities can differ even when the model is correct, and (iii) *MP-hurt* instances that reveal recurring failure modes when MP degrades prediction quality.

6.4.1 MP-Help Cases

In MP-help instances, the MP indicators align with the skills required to solve the problem (e.g., recognizing that the item involves partial sums of an arithmetic sequence). Student responses to these indicators provide item-relevant evidence of mastery, enabling the model to associate correctness with the appropriate underlying proficiency.

6.4.2 Confidence Gap Analysis

We compare an MP-correct low-score case (≈ 0.5) and a high-score case (> 0.7). MPs are similarly well scoped in both, but confidence is lower for the harder item (difficulty 70) than for the easier one

(difficulty 30), suggesting difficulty-driven uncertainty even when MP is adequate.

6.4.3 MP-Hurt Cases

In MP-hurt instances, we find a scope-related failure mode: MP can include non-essential, over-enriched statements (e.g., optional alternative methods) or misaligned with the item requirements reflected in the student's PSP. Such over-enrichment may inflate the perceived skill requirements and lead to an incorrect prediction even when the student answers correctly. This suggests that MP scope and granularity matter for reliable integration, and motivates scope-controlled MP (e.g., separating core vs. enriched indicators) as future work.

7 Conclusion

This work introduced KNOWLEDGE TRACING WITH PROBLEM-SOLVING PROCESS (KT-PSP), a new formulation that incorporates students' solution processes into knowledge tracing, addressing the limitation of outcome-centric KT methods. To support this direction, we constructed KT-PSP-25, the first mathematical KT dataset that provides real-world, OCR-transcribed problem-solving processes along with rich interaction metadata. Building on this foundation, we proposed STATUSKT, a KT framework that employs a teacher-student-teacher LLM pipeline to extract interpretable mathematical proficiency (MP) ratios. These MP ratios serve as meaningful intermediate representations that enrich KT models with process-level information beyond correctness. Experiments on KT-PSP-25 demonstrate that STATUSKT consistently improves prediction performance while providing interpretable proficiency signals. Our study highlights that incorporating the reasoning reflected in students' problem-solving processes is essential for advancing KT toward more accurate, interpretable, and pedagogically meaningful modeling.

8 Limitations

This work may introduce potential risks related to educational bias. Since the dataset is collected from a specific platform and student population, the model may reflect bias that do not generalize across diverse learners. Our dataset is limited in scale, as the number of participating students is relatively small, which may constrain the generalizability of the results. In addition, the domain is restricted to mathematics, and it remains unclear how well the proposed framework would transfer to other subjects or to problem-solving contexts. Because raw student problem-solving processes (PSP) could not be released as images, we relied on OCR to extract the text. This step introduces occasional recognition errors and removes visual reasoning elements, such as diagrams and graphs, which may carry important cognitive signals.

Additionally, the current pipeline relies on GPT-5 for all three LLM stages, which introduces non-trivial computational cost and latency compared to lightweight DLKT baselines. Although indicator extraction is problem-specific and can be pre-computed offline, two LLM calls per student-problem interaction are still required during the evaluation stage. Our primary contribution lies in establishing empirical evidence that process-level signals provide consistent modeling gains across architectures. In future work, we plan to explore distillation and task-specific smaller models to reduce the reliance on large-scale LLM inference.

9 Acknowledgments

This work was partly supported by QANDA (Mathpresso Inc.), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korean Government (MSIT) (No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. RS-2024-00354218), and the Technology Innovation Program(RS-2025-25456760, Development of a humanoid robot specialized in chemical processes based on AI foundation model) funded By the Ministry of Trade, Industry and Resources(MOTIR, Korea). We express special thanks to KAIT GPU project. The ICT at Seoul National University provides research facilities for this study.

References

Ghodai Abdelrahman, Sherif Abdelfattah, Qing Wang, and Yu Lin. 2022. Dbe-kt22: A knowledge tracing dataset based on online student evaluation. *arXiv preprint arXiv:2208.12651*.

Ghodai Abdelrahman and Qing Wang. 2019. Knowledge tracing with sequential key-value memory networks. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 175–184.

Kingma DP Ba J Adam and 1 others. 2014. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412(6).

Youheng Bai, Xueyi Li, Zitao Liu, Yaying Huang, Teng Guo, Mingliang Hou, Feng Xia, and Weiqi Luo. 2025. cskt: Addressing cold-start problem in knowledge tracing via kernel bias and cone attention. *Expert Systems with Applications*, 266:125988.

Indronil Bhattacharjee and Christabel Wayllace. 2025. Cold start problem: An experimental study of knowledge tracing models with new students. *Preprint*, arXiv:2505.21517.

Haw-Shiuan Chang, Hwai-Jung Hsu, Kuan-Ta Chen, and 1 others. 2015. Modeling exercise relationships in e-learning: A unified approach. In *EDM*, pages 532–535.

Jiahao Chen, Zitao Liu, Shuyan Huang, Qiongqiong Liu, and Weiqi Luo. 2023. Improving interpretability of deep sequential knowledge tracing models with question-centric cognitive representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 14196–14204.

Penghe Chen, Yu Lu, Yang Pian, Yan Li, and Yunbo Cao. 2022. Introducing response time into guessing and slipping for cognitive diagnosis. In *International Conference on Artificial Intelligence in Education*, pages 320–324. Springer.

Barbara Chiu, Christopher Randles, and Stefan Irby. 2022. Analyzing student problem-solving with match. In *Frontiers in Education*, volume 6, page 769042. Frontiers Media SA.

Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. 2020a. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the seventh ACM conference on learning@ scale*, pages 341–344.

Youngduck Choi, Youngnam Lee, Dongmin Shin, Junghyun Cho, Seoyon Park, Seewoo Lee, Jineon Baek, Chan Bae, Byungsoo Kim, and Jaewe Heo. 2020b. Ednet: A large-scale hierarchical dataset in education. In *International conference on artificial intelligence in education*, pages 69–73. Springer.

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

Ibrahim AH El-Shara, Ahmad AS Tabieh, and Sahar YA Abu Helu. 2025. The effect of using matgpt on mathematical proficiency among undergraduate students. *International Journal of Information and Education Technology*, 15(4).

Mingyu Feng, Neil Heffernan, and Kenneth Koedinger. 2009. Addressing the assessment challenge with an online system that tutors as it assesses. *User modeling and user-adapted interaction*, 19(3):243–266.

Bradford Findell, Jane Swafford, and Jeremy Kilpatrick. 2001. *Adding it up: Helping children learn mathematics*. National Academies Press.

Aritra Ghosh, Neil Heffernan, and Andrew S Lan. 2020. Context-aware attentive knowledge tracing. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2330–2339.

Bert F Green Jr. 1951. A general solution for the latent class model of latent structure analysis. *Psychometrika*, 16(2):151–166.

Teng Guo, Yu Qin, Yubin Xia, Mingliang Hou, Zitao Liu, Feng Xia, and Weiqi Luo. 2025. Enhancing knowledge tracing through decoupling cognitive pattern from error-prone data. In *Proceedings of the ACM on Web Conference 2025*, pages 5108–5116.

Xiaopeng Guo, Zhijie Huang, Jie Gao, Mingyu Shang, Maojing Shu, and Jun Sun. 2021. Enhancing knowledge tracing via adversarial training. In *Proceedings of the 29th ACM international conference on multimedia*, pages 367–375.

Yuxiang Guo, Shuanghong Shen, Qi Liu, Zhenya Huang, Linbo Zhu, Yu Su, and Enhong Chen. 2024. 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*, CIKM '24, page 727–736, New York, NY, USA. Association for Computing Machinery.

Tao Huang, Shengze Hu, Huali Yang, Jing Geng, Zhifei Li, Zhuoran Xu, and Xinjia Ou. 2024. Response speed enhanced fine-grained knowledge tracing: A multi-task learning perspective. *Expert Systems with Applications*, 238:122107.

KDD 2010. 2010. Kdd2010. https://kdd.org/kdd-cup/view/kdd-cup-2010-student-performance-evaluation. Accessed: 2010.

Dohee Kim, Unggi Lee, Sookbun Lee, Jiyeong Bae, Taekyung Ahn, Jaekwon Park, Gunho Lee, and Hyeoncheol Kim. 2025. Es-kt-24: A multimodal knowledge tracing benchmark dataset with educational game playing video and synthetic text generation. In *International Conference on Intelligent Tutoring Systems*, pages 259–273. Springer.

Erica Kleinman, Murtuza Shergadwala, Zhaoqing Teng, Jennifer Villareale, Andy Bryant, Jichen Zhu, and Magy Seif El-Nasr. 2022. Analyzing students' problem-solving sequences: A human-in-the-loop approach. *Journal of learning analytics*, 9(2):138–160.

Xueyi Li, Youheng Bai, Teng Guo, Zitao Liu, Yaying Huang, Xiangyu Zhao, Feng Xia, Weiqi Luo, and Jian Weng. 2024. Enhancing length generalization for attention based knowledge tracing models with linear biases. In *Proceedings of the thirty-third international joint conference on artificial intelligence (IJCAI-24)*, pages 5918–5926.

Guimei Liu, Huijing Zhan, and Jung-jae Kim. 2024. Question difficulty consistent knowledge tracing. In *Proceedings of the ACM Web Conference 2024*, pages 4239–4248.

Zitao Liu, Teng Guo, Qianru Liang, Mingliang Hou, Bojun Zhan, Jiliang Tang, Weiqi Luo, and Jian Weng. 2025. Deep learning based knowledge tracing: A review, a tool and empirical studies. *IEEE Transactions on Knowledge and Data Engineering*.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. 2023a. simplekt: a simple but tough-to-beat baseline for knowledge tracing. *arXiv preprint arXiv:2302.06881*.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, Jiliang Tang, and Weiqi Luo. 2022. pykt: A python library to benchmark deep learning based knowledge tracing models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zitao Liu, Qiongqiong Liu, Teng Guo, Jiahao Chen, Shuyan Huang, Xiangyu Zhao, Jiliang Tang, Weiqi Luo, and Jian Weng. 2023b. Xes3g5m: A knowledge tracing benchmark dataset with auxiliary information. *Advances in Neural Information Processing Systems*, 36:32958–32970.

Koki Nagatani, Qian Zhang, Masahiro Sato, Yan-Ying Chen, Francine Chen, and Tomoko Ohkuma. 2019. Augmenting knowledge tracing by considering forgetting behavior. In *The world wide web conference*, pages 3101–3107.

Hiromi Nakagawa, Yusuke Iwasawa, and Yutaka Matsuo. 2019. Graph-based knowledge tracing: modeling student proficiency using graph neural network. In *IEEE/WIC/aCM international conference on web intelligence*, pages 156–163.

OpenAI. 2025. Gpt-5 system card. https://cdn.openai.com/gpt-5-system-card.pdf. Accessed: 2025-09-24.

Shalini Pandey and George Karypis. 2019. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*.

Zachary A Pardos, Ryan SJD Baker, Maria OCZ San Pedro, Sujith M Gowda, and Supreeth M Gowda. 2014. Affective states and state tests: Investigating how affect and engagement during the school year predict end-of-year learning outcomes. *Journal of learning Analytics*, 1(1):107–128.

Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. 2015. Deep knowledge tracing. *Advances in neural information processing systems*, 28.

Stefan Slater and Ryan S. Baker. 2018. Degree of error in Bayesian knowledge tracing estimates from differences in sample sizes. *Behaviormetrika*, 45(2):475–493.

Yu Su, Zeyu Cheng, Pengfei Luo, Jinze Wu, Lei Zhang, Qi Liu, and Shijin Wang. 2021. Time-and-concept enhanced deep multidimensional item response theory for interpretable knowledge tracing. *Knowledge-Based Systems*, 218:106819.

Peter Sullivan. 2011. Using the proficiencies from the australian mathematics curriculum to enrich mathematics teaching and assessment. *Australian Education Review*, (59).

Paul Tschisgale, Marcus Kubsch, Peter Wulff, Stefan Petersen, and Knut Neumann. 2025. Exploring the sequential structure of students' physics problem-solving approaches using process mining and sequence analysis. *Physical Review Physics Education Research*, 21(1):010111.

Fidele Ukobizaba, Gabriel Nizeyimana, and Angel Mukuka. 2021. Assessment strategies for enhancing students' mathematical problem-solving skills: A review of literature. *Eurasia Journal of Mathematics, Science and Technology Education*, 17(3).

Yutao Wang and Neil T Heffernan. 2012. Leveraging first response time into the knowledge tracing model. *International Educational Data Mining Society*.

Yang Yang, Jian Shen, Yanru Qu, Yunfei Liu, Kerong Wang, Yaoming Zhu, Weinan Zhang, and Yong Yu. 2020. Gikt: A graph-based interaction model for knowledge tracing. *Preprint*, arXiv:2009.05991.

Chun-Kit Yeung. 2019. Deep-irt: Make deep learning based knowledge tracing explainable using item response theory. *arXiv preprint arXiv:1904.11738*.

Chun-Kit Yeung and Dit-Yan Yeung. 2018. Addressing two problems in deep knowledge tracing via prediction-consistent regularization. In *Proceedings of the fifth annual ACM conference on learning at scale*, pages 1–10.

Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774.

Yiyun Zhou, Wenkang Han, and Jingyuan Chen. 2025. Revisiting applicable and comprehensive knowledge tracing in large-scale data. *arXiv preprint arXiv:2501.14256*.

A Baseline Models

- **DKT** (Piech et al., 2015): The first RNN-based knowledge tracing model capturing temporal learning patterns.
- **DKT+** (Yeung and Yeung, 2018): Adds regularization to DKT to mitigate overfitting and improve interpretability.
- **DKVMN** (Zhang et al., 2017): Introduces a dynamic key-value memory network to explicitly model concept-level knowledge states.
- **SKVMN** (Abdelrahman and Wang, 2019): Enhances DKVMN by disentangling student and skill representations for better interpretability.
- **SAKT** (Pandey and Karypis, 2019): Employs self-attention to identify the most relevant past interactions efficiently.
- **AKT** (Ghosh et al., 2020): Incorporates distance-aware exponential decay to model learning and forgetting behaviors.
- **SimpleKT** (Liu et al., 2023a): Simplifies the attention architecture for computational efficiency while maintaining accuracy.
- **StableKT** (Li et al., 2024): Improves model stability and generalization on long student interaction sequences.
- **RobustKT** (Guo et al., 2025): Enhances robustness against noisy or incomplete student data through robust optimization techniques.

B Dataset Comparison

Table 5 provides a summary of widely utilized knowledge tracing (KT) datasets and the types of information they encompass. Earlier benchmarks, predominantly include correctness logs and a limited set of contextual attributes. More recent datasets offer enhanced metadata, such as question text or type, but they still lack explicit documentation of the methods by which students arrive at their answers.

In contrast, KT-PSP-25 incorporates students' problem-solving processes (PSP) as textual traces alongside conventional KT attributes. This process-level information enables models not only to predict future performance but also to analyze the reasons behind students' successes or failures, thereby supporting more interpretable and process-aware KT research.

Dataset	Question Difficulty	Timestamp	Question Text	Question Type	Student's PSP
ASSISTments2009	✓	✗	✗	✗	✗
ASSISTments2014	✓	✗	✗	✗	✗
Junyi2015	✓	✓	✗	✗	✗
KDDcup2010	✗	✓	✗	✗	✗
EdNet	✗	✓	✗	✗	✗
DBE-KT22	✓	✓	✓	✗	✗
XES3G5M	✗	✓	✓	✓	✗
ES-KT-24	✗	✓	✓	✓	✗
KT-PSP-25(**Ours**)	✓	✓	✓	✓	✓

Table 5: **Comparison of educational datasets commonly used in KT.** While prior datasets do not include students' PSP, our KT-PSP-25 is the first to offer process-level textual traces enabling process-aware KT modeling.

C Training Details

All experiments were conducted using the pyKT library (Liu et al., 2022), which is a PyTorch-based framework for knowledge tracing. We searched the learning rate from $[5 \times 10^{-3},\ 1 \times 10^{-3},\ 5 \times 10^{-4},\ 1 \times 10^{-4}]$, dropout from $[0.5,\ 0.3,\ 0.1,\ 0.05]$, and α from $[0.01,\ 0.25,\ 1.0,\ 1.5,\ 2.0]$. A fixed random seed (42) was used for the reproducibility. All the training was conducted using a single NVIDIA RTX 3090 GPU.

For the LLM-based components in the pipeline, we used the gpt-5 model with the temperature set to 0.0 and top-p set to 1.0 to ensure deterministic behavior. We did not impose an explicit maximum token limit, allowing the model to generate complete JSON outputs as required by the prompt.

The detailed model configurations for each baseline model are summarized in Table 6

Model	FFN Dim	Heads	Blocks	Additional Architecture Details
DKT	–	–	–	–
DKT+	–	–	–	λ_r=0.01, λ_{w1}=0.003, λ_{w2}=3.0
DKVMN	–	–	–	memory size=50
SKVMN	–	–	–	memory size=50
SAKT	256	8	1	–
SAINT	256	8	4	–
AKT	512	8	4	–
simpleKT	256	4	2	start=50, num_layers=2, final_fc_dim=256
stableKT	256	8	4	num_layers=2, r=0.5, γ=0.7
robustKT	512	8	4	k_s=5

Table 6: Model configurations for each baseline architecture.

D OCR Pipeline Details

D.1 Prompts

The prompts used in our dataset creation process were as follows:

- Figure 5: A prompt for GPT-based OCR to convert students' handwritten problem-solving process into text.
- Figure 6: A prompt for refining GPT-based OCR outputs to improve transcription quality.

D.2 OCR Quality Evaluation

To assess the reliability of our OCR pipeline, we manually inspected a randomly selected subset of 100 handwritten PSP samples from the KT-PSP-25. Each OCR output was categorized as either *usable* (i.e., it retained the original mathematical meaning) or *unusable* (i.e., it exhibited significant distortions such as missing operators, collapsed fractions, or omitted lines). According to this criterion, 85% of the sampled outputs were deemed usable, whereas the remaining 15% contained major transcription errors. The inspection was conducted by trained annotators with expertise in mathematics and AI, following a short written guideline describing these criteria.

In our current experiments, we did not explicitly correct or filter unusable OCR cases, indicating that some of the downstream noise in MP extraction and KT prediction likely originated from the OCR failures. Nevertheless, we observed that most unusable cases resulted from structural breakdowns in complex handwritten expressions rather than widespread systematic misrecognition. Consequently, we regard OCR noise as a moderate but manageable limitation of our pipeline, and we defer more systematic robustness studies and denoising strategies to address this issue in future research.

E Mathematical Proficiency Extraction Details

E.1 Mathematical Proficiency Human Evaluation

To evaluate the reliability of the automatically generated MP indicators, a human assessment was conducted on 50 randomly selected items, comprising 25 correct and 25 incorrect responses. In total, annotators evaluated 590 indicators. Each indicator was assessed using a three-level rubric:

- 0 — unnecessary or irrelevant to solving the problem,
- 1 — conceptually valid but misaligned with the student's final answer,
- 2 — appropriate and consistent with both the solution and the answer.

Overall, 55.8% of indicators received a score of 2, indicating that slightly more than half of the generated indicators accurately captured students' reasoning. However, 35.8% were deemed partially valid (score = 1), typically reflecting situations where the reasoning statement itself was correct but did not correspond to the student's final (often incorrect) response. A smaller fraction (8.3%) consisted of redundant or clearly unnecessary indicators (score = 0).

When analyzed by proficiency dimension, notable variation was observed: indicators related to conceptual understanding (CU; mean = 1.60) and procedural fluency (PF; mean = 1.59) were generally reliable, whereas adaptive reasoning (AR) achieved the lowest average score (1.13), with a comparatively large share of unnecessary statements. This suggests that generating meta-justifications and verification steps remains challenging for current LLMs. Further comparison of indicators between correct and incorrect responses revealed that for student responses marked as correct, 67.6% of indicators received the maximum score, compared to only 44.6% for incorrect responses. Conversely, partially correct indicators (score = 1) were substantially more frequent in incorrect cases (44.9% vs. 26.1%). These results indicate that MP indicators convey meaningful information about the quality and stability of students' reasoning, beyond merely their final answers.

E.2 Prompts

The prompts used in our StatusKT for extracting the mathematical proficiency(MP) ratio are as follows:

- Figure 7: A prompt to extract MP indicators from problem statement and curricular unit name.
- Figure 8: A prompt for student LLM, which simulates how a student would respond to each of the generated indicators.

- Figure 9: A prompt to evaluate whether each response generated by student LLM satisfies the intent of its corresponding indicator.

Raw Prompt:
"You are a Korean Mathematical OCR Specialist with logical sequencing capability.
Extract textual content from Korean student math work, then reorder it into logical mathematical sequence.
Ignore all visual elements (graphs, diagrams, shapes). Output in LaTeX format with proper mathematical flow."

Prompt Categories:
- Role & Identity: Korean Mathematical OCR Specialist + Logic Sequencer
- Output Format: LaTeX Mathematical Text (Logically Ordered)
- Cognitive Bias Lever: Neutral
- Creativity/Fidelity Balance: 100% Fidelity + Logic Enhancement

User Settings:
- Korean Language Processing: true
- Visual Element Filtering: true (IGNORE all visuals)
- Fraction Normalization: true (vertical → \\frac{}{})
- Extract Text Only: true
- LaTeX Output: true
- Logic Sequencing: true

Core Instructions:
Meta-Cognitive Pre-Check:
1. "What mathematical text do I see?"
2. "What visual elements must I ignore?"
3. "How do I convert fractions to LaTeX?"
4. "What is the logical mathematical sequence here?"

Two-Phase Processing
Phase1: Raw Extraction
- **IGNORE**: graphs, figures, diagrams, arrows, connecting lines, numbers that consists figure
- **EXTRACT**: Korean text, formulas, numbers, mathematical symbols
- **CONVERT**: vertical fractions → \\frac{numerator}{denominator}
- **SCAN**: ALL content regardless of position

Phase2: Logic Sequencing
- ANALYZE: analyze given problem and raw extracted solving traces
- IDENTIFY: problem statement vs. solution steps
- DETECT: calculation flow (which leads to which)
- REORDER: arrange in logical mathematical sequence
- **DO NOT** change the extracted text

LaTeX Formatting:
- fraction: \\frac{3}{4}
- exponent: x^{2}
- square root: \\sqrt{x}
- parentheses: ( )

FINAL INSTRUCTION:
First, extract all mathematical text content. Then, reorder into logical mathematical sequence.

Output ONLY the extracted mathematical content in LaTeX format.
No commentary, no process descriptions, no given problem, only the components in given image.

PRESERVATION MANDATE: You are an OCR system, NOT a math tutor. Your job is to organize student work, not to correct it. Preserve all original content exactly as the student wrote it.

Once again, **Never output the given problem.**

Figure 5: Prompt used for GPT-based OCR to convert students' handwritten problem-solving processes into text.

```
system_msg = (
    "You are a LaTeX OCR fixer. "
    "Your sole job is to correct OCR-induced errors in LaTeX while preserving meaning."
)

guardrails = (
    "You are a LaTeX OCR fixer.\n\n"
    "STRICT RULES:\n"
    "1) Fix ONLY OCR-induced errors in the provided LaTeX. Do not change mathematical meaning
    ↪  beyond what is necessary to correct OCR mistakes.\n"
    "2) Do NOT add explanations, comments, opinions, or extra text. Output MUST be only the
    ↪  corrected LaTeX code.\n"
    "3) Use the provided references (problem statement, choices, model answer) strictly to
    ↪  resolve ambiguities and to choose the correct symbols/operators/numbers. Prefer the
    ↪  minimal edit that matches the references.\n"
    "4) **Do NOT** introduce new steps, reorder lines, simplify, expand, compute results, or
    ↪  rename variables unless correcting an OCR error that conflicts with the references.\n"
    "5) Preserve structure and line breaks of the input LaTeX unless required to fix syntax or
    ↪  OCR mistakes. Keep environments (inline/display) as-is where possible.\n"
    "6) **Ensure** syntactic validity: balanced braces, valid commands, proper math mode,
    ↪  correct subscripts/superscripts, properly paired \\left ... \\right, etc.\n"
    "7) If the input is already correct, return it unchanged.\n"
    "8) If the input is irrecoverably ambiguous, return the minimally fixed version that
    ↪  compiles, without adding any new content.\n"
)

rails = (
    "OUTPUT CONTRACT:\n"
    "- Return **ONLY** a single fenced code block labeled 'latex' containing the corrected
    ↪  LaTeX.\n"
    "- No surrounding prose, no markdown outside the code block, no comments.\n"
)

inputs = {
        "problem_text": problem,
        "solution_explanation": solution,
        "ocr_latex": ocr_latex,
}

user_msg = (
    ((user_prompt.strip() + "\n\n") if user_prompt else "") +
    (guardrails.strip() + "\n\n") +
    rails + "\n\n" +
    "INPUTS(JSON):\n" + json.dumps(inputs, ensure_ascii=False)
)
```

Figure 6: Prompt used for refining the GPT-based OCR outputs to improve the accuracy and consistency of transcribed student solutions in our KT-PSP-25.

```
You are Teacher GPT.
Your task is to analyze a given math Problem and its Unit name, and then generate a step-by-step
set of indicators that describe the process a student should ideally follow to solve the problem.

###Guidelines:
- The indicators must be organized into the four categories of Mathematical Proficiency:
  - Conceptual Understanding (CU)
  - Procedural Fluency (PF)
  - Strategic Competence (SC)
  - Adaptive Reasoning (AR)

- However, instead of just listing general skills, write the indicators as concrete *steps*
that a student would naturally take while solving the given problem.
- Each indicator should be prefixed with its category code (e.g., "CU1", "PF1", "SC2", "AR3").
- The order of the indicators should roughly follow the logical order of problem solving
(from initial understanding → strategy selection → execution → justification).

- Output format must be a JSON dictionary:
{
  "mathematical_proficiency_indicators": [
    "CU1": "...",
    "SC1": "...",
    "CU2": "...",
    ...
  ]
}

---
### One-shot Example

**Input**
Problem: Solve the differential equation $(\frac{dy}{dx} = 2x)$ with initial condition $(y(0)=1)$."
Unit: Differential Equations

**Output**
{
  "mathematical\_proficiency\_indicators": [
    {"CU1": "Determine the type and order of this equation"},
    {"SC1": "Rewrite the equation in an easier way"},
    {"CU2": "Write the mathematical idea you need to solve this equation"},
    {"CU3": "Give an example of how this equation will be applied in real life"},
    {"CU4": "Find another differential equation whose solution steps are similar"},
    {"SC2": "Sort the necessary data and ignore the redundant ones"},
    {"PF2": "Predict a solution"},
    {"CU5": "Show the steps for solving the equation using a table, a figure and a diagram"},
    {"PF1": "Summarize the steps in the solution"},
    {"PF3": "Write a suitable algorithm to solve this equation"},
    {"SC3": "Identify any special numerical cases used by this equation to generalize the
solution"},
    {"AR1": "Describe your solution in general"},
    {"AR2": "Based on your knowledge of differential equations, interpret your solution"},
    {"AR3": "According to your solution, draw the conclusions"}
  ]
}
```

Problem (in Korean): {**Problem_text**}
{**problem_option_string**}
Unit (in Korean): {**curriculum_theme_title**}

Figure 7: Prompt used for extracting the MP indicators from the given problem in StatusKT. Prompt inputs are **boldfaced**.

```
You are Student GPT.
You will receive:
1. A math problem statement.
2. A set of indicators generated by Teacher GPT.
3. A student's written solution attempt (from OCR).

Your task:
    - Pretend you are the student who wrote the solution.
    - For each indicator, provide an answer based **only on the student's written solution**.
    - If the student's solution clearly contains the relevant information, restate it as the
answer.
    - If a step is missing but can be reasonably inferred (e.g., a basic algebraic manipulation
or obvious arithmetic), you may state it as: "Not written, but likely ...".
    - Keep the student's mistakes. **Do not** correct them.
    - If step looks incomplete or skipped, you can imagine that step and answer to indicator.
    - If there is no evidence in the solution for an indicator, answer with: "I don't know"

**Output format**
Return your answers as a dictionary, and indicators should be written in Korean.
Output:
{
    "CU1": "...",
    "SC1": "...",
    "CU2": "...",
    ...
}


---

### One-shot Example

Input Indicators:
{
    "CU1": "Determine the type and order of this equation",
    "SC1": "Rewrite the equation in a simpler form",
    "AD1": "Identify the conditions required to solve the equation",
    "PF1": "Compute the values that satisfy the conditions"
}

Question: Find the value(s) of y that make the following expression equal to 0.
                      y^2 + 3y + 2

My solving process (OCR):
    `y^2 + 3y' + 2y = 0`
    `(r+1)(r+2)=0`

My answer: -1, -2

Output:
{
    "CU1": "This is a quadratic equation.",
    "SC1": "Rewrite the characteristic polynomial as (r+1)(r+2)=0.",
    "AD1": "If one of the multiplied factors is zero, the result becomes zero.",
    "PF1": "The values that satisfy the condition are -1 and -2.."
}
```

Input Indicators: {**indicator_text**}

Problem (in Korean): {**problem**}
{**problem_option_string**}

My solving process (OCR):{**student_solving_trace**}

My answer: {**solution_answer_sets**}

Figure 8: Prompt used for generating responses corresponding to each MP indicator in StatusKT. Prompt inputs are **boldfaced**.

```
You are Teacher GPT. Your task is to evaluate a student's responses (answer_indicate)
against the reference mathematical proficiency indicators (mathematical_proficiency_indicators).

## Evaluation Rules
1. For each indicator:
   - If the student's response is **"I don't know"**, assign 0.
   - If the student's response is **"Not written, but likely ..."**, treat it
as the student's actual answer and evaluate normally.
   - If the response does not match or is irrelevant to the indicator, assign 0.
   - If the response matches the indicator's intent and shows correct reasoning/application,
assign 1.
2. Output strictly in JSON format, with indicator keys mapped to 0 or 1.
3. Ensure that every indicator is carefully evaluated without skipping or overlooking any of them.

## Input
Problem:
{problem_text}

mathematical_proficiency_indicators:
{mathematical_proficiency_indicators JSON}

answer_indicate:
{answer_indicate JSON}

## Output
Provide the evaluation result in the following JSON format:
{
   "CU1": 0 or 1, "CU2": 0 or 1, "SC1": 0 or 1, "SC2": 0 or 1, "PF1": 0 or 1, "PF2": 0 or 1, "AR1": 0 or 1, "PF3": 0 or 1, "PF4": 0 or 1, "AR2": 0 or 1
}

## Input Example
Problem: For the rational function $y=\\dfrac{2x-3}{2x+5}$, how many points on its graph have both
$x$- and $y$-coordinates as integers?
Options: [{"index":1,"text":"$1$"}, {"index":2,"text":"$2$"}, {"index":3,"text":"$3$"},
{"index":4,"text":"$4$"}, {"index":5,"text":"$5$"}]

mathematical_proficiency_indicators:[
    {"CU1": "Interpret what the problem is asking, and recognize that it is about finding points on the graph of the rational function
y = (2x - 3) / (2x + 5) whose x- and y-values are both integers."},
    {"CU2": "Identify the domain restriction 2x + 5 is not 0, and observe that when x is an integer, 2x + 5 is always odd."},
    {"SC1": "Choose a strategy to rewrite the equation in a form that makes the integer condition more explicit, such as expressing it in terms
of y - 1."},
    {"PF1": "Transform y = (2x - 3)/(2x + 5) into y - 1 = [(2x - 3) - (2x + 5)]/(2x + 5) =
-8/(2x + 5)."},
    {"SC2": "Since y must be an integer, -8/(2x + 5) must be an integer; thus reinterpret this as the divisibility condition 2x + 5 | 8."},
    {"AR1": "Use the fact that 2x + 5 is odd to restrict the candidates to the odd divisors of 8."},
    {"PF2": "List the possible denominators: 2x + 5 \in {1, -1}."},
    {"PF3": "Solve for x for each candidate: 2x + 5 = 1 => x = -2; 2x + 5 = -1 => x = -3."},
    {"PF4": "For each x, compute y using y = 1 - 8/(2x + 5): for x = -2 => y = -7; for x = -3 => y = 9."},
    {"PF5": "Verify that the points (-2, -7) and (-3, 9) satisfy the original equation y = (2x - 3)/(2x + 5)."},
   {"AR2": "Provide reasoning that only +±1 or -1 can occur, since all odd divisors of 8 have been fully checked and no others are possible."},
    {"SC3": "Alternative check: assuming y is not equal to 1, set x = -(5y + 3)/(2(y - 1)). Let d = y - 1. Then x = -5/2 - 4/d, and
x is an integer only when d = ±8, confirming the two solutions (-2, -7) and (-3, 9)."},
    {"CU3": "Count the integer lattice points obtained and select the corresponding choice from
the answer options."}
]

answer_indicate:[
    {"CU1": "Although not written explicitly, by rewriting y = (2x - 3)/(2x + 5) as y = -8/(2x + 5) + 1 and listing possible values of 2x + 5
to find integer pairs (x, y), it appears the student recognized the task as identifying integer lattice points."},
   {"CU2": "The student did not mention the condition 2x + 5 is not equal to 0 or the observation that 2x + 5 must be odd when x is an integer.
Instead, they listed all divisors of 8 (1, 2, 4, 8, -1, -2, -4, -8)."},
    {"SC1": "They rewrote y = (2x - 3)/(2x + 5) as y = -8/(2x + 5) + 1."},
    {"PF1": "Although intermediate algebra steps were omitted, the final expression y - 1 = -8/(2x + 5) was obtained."},
    {"SC2": "They interpreted the requirement that -8/(2x + 5) must be an integer by considering all divisors of 8, listing
2x + 5 = 1, 2, 4, 8, -1, -2, -4, -8."},
    {"AR1": "They did not use the constraint that 2x + 5 must be odd, which would reduce the candidates to the odd divisors only."},
    {"PF2": "They listed 2x + 5 \in {1, 2, 4, 8, -1, -2, -4, -8} as possible candidates."},
    {"PF3": "They solved only some candidates: 2x + 5 = 1 => x = -2, and 2x + 5 = -1 => x = -3. The remaining cases were left incomplete
or not shown."},
    {"PF4": "For x = -2 and x = -3, the y-values were left blank; but likely x = -2 => y = -7, and x = -3 => y = 9."},
    {"PF5": "I don't know."},
    {"AR2": "I don't know."},
    {"SC3": "I don't know."},
    {"CU3": "The student eventually selected choice (2)."}

]

Output:
{
    "CU1": 1, "CU2": 0, "SC1": 1, "PF1": 1, "SC2": 1, "AR1": 0, "PF2": 1, "PF3": 1, "PF4": 1, "PF5": 0, "AR2": 0, "SC3": 0, "CU3": 1
}
```

Problem (in Korean): {**problem**}
{**problem_option_string**}

Mathematical Proficiency Indicators:
{**indicator_text**}

Answer Indicate: {**answer_indicator_text**}

Figure 9: Prompt used for evaluates the appropriateness of each generated response for its corresponding indicator in STATUSKT. Prompt inputs are **boldfaced**.