

Tracing Mathematical Proficiency Through Problem-Solving Processes

Jungyang Park^{1,2*} Suho Kang^{3*} Jaewoo Park¹
 Jaehong Kim² Jaewoo Shin² Seonjoon Park² Youngjae Yu³

¹Yonsei University ²Mathpresso ³Seoul 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 the KT-PSP. Building on this, we present StatusKT, a KT framework that employs a teacher-student-teacher three-stage LLM pipeline to extract students’ MP as intermediate signals. 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.

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). However, directly obtaining the true knowledge state remains inherently challenging. In practice, KT addresses this limitation by exploiting learners’ historical interaction data to predict the likelihood of correctness in subsequent exercises, 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 capture long-term dependencies. 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), student attempt history (Luo et al., 2022), and knowledge concept structure (Su et al., 2021), thereby enhancing their performance and interpretability.

Although KT has advanced in various ways, most approaches rely on outcome-based signals, such as answer correctness, knowledge concepts, problem-response time, and question difficulty (Wang and Heffernan, 2012; Tschisgale et al., 2025; Yeung, 2019; Liu et al., 2024). However, such signals often fail to reflect students’ understanding. In real-world educational contexts, as illustrated in Figure 1 (b), teachers evaluate understanding through the problem-solving process rather than outcomes alone (Chiu et al., 2022; Tschisgale et al., 2025; Kleinman et al., 2022). Nevertheless, despite the importance of this process-oriented perspective, 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 into the interaction sequence. To facilitate research on KT-PSP, we further in-

* Equal Contribution.

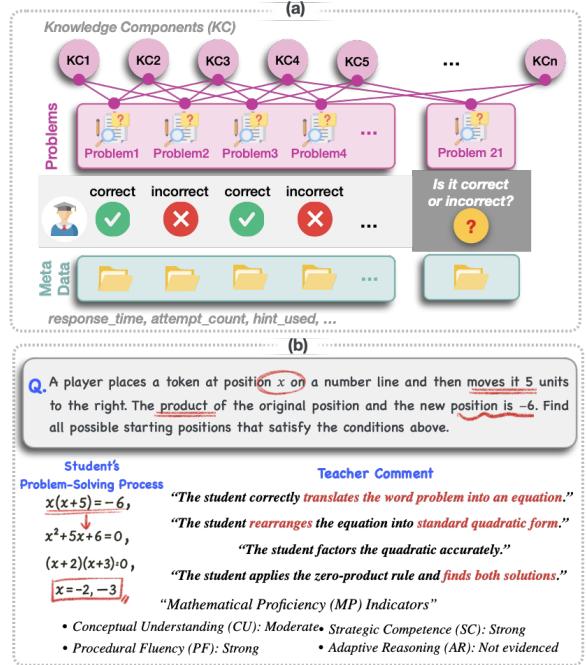


Figure 1: (a) Conventional KT utilizes limited metadata, lacking interpretability. (b) By incorporating problems, students reasoning, and teacher comments, StatusKT captures granular proficiency like a teacher.

introduce KT-PSP-25, a novel dataset that contains real-world problem-solving processes for each student-problem interaction. While the original handwritten images cannot be released due to privacy and policy constraints, we provide OCR transcriptions of the solution processes. Alongside these process-level texts, the dataset includes rich problem metadata (knowledge concepts, difficulty, question text, solution text) and interaction attributes (selected answer, duration, correctness)

Furthermore, we introduce STATUSKT, a KT framework that incorporates students’ mathematical proficiency (MP)—defined in Findell et al. (2001) as comprising five components—extracted from their problem-solving processes as an intermediate signal for modeling student knowledge, leveraging this rich information as shown in Figure 1 (b). Inspired by El-Shara et al. (2025), who assessed four observable MP dimensions using a structured, step-based evaluation, STATUSKT adopts a similar teacher-student-teacher LLM pipeline to extract MP signals. Specifically, a teacher LLM derives problem-specific MP indicators; a student LLM responds to them based on the student’s solution process; and a teacher LLM evaluates these responses to produce problem-level MP signals. These MP signals are then integrated into a KT backbone as auxiliary inputs, enabling the model to incorporate

proficiency-level information, thereby enhancing both its predictive accuracy and interpretability.

Our experiments on KT-PSP-25 demonstrate that STATUSKT consistently improves the performance of strong DLKT baselines across multiple metrics. In addition to higher predictive accuracy, STATUSKT provides interpretable proficiency signals that offer fine-grained insights into students’ learning progress. Our contributions are as follows:

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

2 Related Works

2.1 Knowledge Tracing

KT has played a crucial role in ITS by enabling accurate assessment of students’ progressive knowledge states. A well-studied approach in KT relies on students’ learning interaction data to infer and predict their understanding of specific concepts in an auto-regressive manner over time (Piech et al., 2015; Yeung and Yeung, 2018; Nagatani et al., 2019; Guo et al., 2021; Zhou et al., 2025). Other models employ memory-augmented neural networks for concept embedding, 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). Moreover, graph neural network (GNN) based approaches (Nakagawa et al., 2019; Yang et al., 2020) model relationships between questions and KCs, aggregating current embeddings with historical states and performing graph-structured updates—often over implicit/learned edges—to yield time-series, node-level

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 1: Comparison of educational datasets commonly used in KT (ASSISTments (Feng et al., 2009; Par-dos et al., 2014), Junyi Academy (Chang et al., 2015), KDD2010, EdNet (Choi et al., 2020b), DBE-KT22 (Ab-delrahman et al., 2022), XES3G5M (Liu et al., 2023b), ES-KT-24 (Kim et al., 2025)). While prior datasets primarily provide correctness, timestamps or problem metadata, none include students’ problem-solving process(PSP). Our KT-PSP-25 is the first to offer process-level textual traces, enabling process-aware KT modeling.

predictions. Attention-based approaches in KT have shown strong potential in capturing the relationship between students’ past interactions and future problems (Pandey and Karypis, 2019; Choi et al., 2020a; Ghosh et al., 2020; Liu et al., 2023a; Li et al., 2024). While effective, these models can be sensitive to noisy or confounded behavioral signals. For robustness, it is crucial to model the key factors that shape students’ knowledge states. Guo et al. (2025) decouple stable cognitive patterns from error-prone factors that induce anomalous outcomes, such as slips and incidental question-level mistakes.

However, key determinants remain underexplored, most notably students’ problem-solving processes and their underlying mathematical proficiency, which likely mediate the mapping from interaction traces to knowledge states. Accordingly, we propose a method that leverages fine-grained signals from students’ problem-solving processes to infer their mathematical proficiency (Findell et al., 2001; Sullivan, 2011), thereby enhancing knowledge tracing.

2.2 Knowledge Tracing Dataset

The ASSISTments datasets (Feng et al., 2009; Par-dos et al., 2014) served as early large-scale benchmarks for DLKT models and laid the foundation for research in knowledge tracing. Following this, several datasets, such as Junyi 2015 (Chang et al., 2015), were introduced, leveraging online learning

activity logs. In parallel, growing attention was directed toward evaluating student performance beyond simple correctness prediction, particularly in assessing how students apply their knowledge to solve problems. This interest culminated in data mining competitions such as the KDD Cup 2010 (KDD 2010, 2010), which focused on evaluating students’ problem-solving abilities using rich learning sequence data. However, these early datasets typically provided only minimal auxiliary information and consisted primarily of simple question-answer sequences, limiting their ability to capture deeper semantic or behavioral aspects of learning. Over time, knowledge tracing has expanded beyond mathematics to other subject areas, including English, computer science, engineering, arts, business, and early childhood education (Choi et al., 2020b; Abdelrahman et al., 2022; Kim et al., 2025). This trend highlights the growing need for KT datasets and models that support richer multimodal and cross-domain learning scenarios. A recent approach by Liu et al. (2023b) incorporates rich auxiliary information, such as question content, answer explanations, and hierarchical KC structures, to support more comprehensive evaluation of DLKT models.

However, students’ problem-solving processes remain underexplored, despite their importance, particularly in the context of math problems, as they are difficult to collect and analyze at scale. To address this gap, we construct the first dataset that captures students’ problem-solving processes.

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}

$$\mathcal{S}_{conv} = \{(q_1, c_1, r_1), (q_2, c_2, r_2), \dots, (q_t, c_t, r_t)\},$$

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 $P(r_{t+1} = 1 | q_{t+1}, c_{t+1}, \mathcal{S}_{conv})$.

However, relying solely on response correctness (r_t) creates a simplified proxy for students’ under-

standing, 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 process, 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). Additionally, interaction features such as hint usage or the number of attempts have been employed to infer a student's struggle level (Rachatasumrit and Koedinger, 2021; Xu et al., 2023), . Although these approaches enrich the context, they largely rely on metadata or static attributes, treating the actual problem-solving procedure as a black box.

In contrast, in programming education domain, Ross et al. (2025) demonstrated that leveraging students' editing traces provides richer signals of reasoning and learning behaviors. However, in the mathematical domain, where the problem-solving process is as critical as the final answer, process-level modeling remains unexplored.

Therefore, we propose KNOWLEDGE TRACING WITH PROBLEM-SOLVING PROCESS (KT-PSP), which extends the traditional KT paradigm by incorporating students' problem-solving processes into the interaction history sequence \mathcal{S}_{new}

$$\mathcal{S}_{new} = \{(q_1, c_1, r_1, p_1), (q_2, c_2, r_2, p_2), \dots, (q_t, c_t, r_t, p_t)\},$$

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

4 Dataset

4.1 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; (2) discard problems with missing textual content. The resulting dataset 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 (student ID, selected answer, duration, problem-solving process, and final correctness).

Beyond these structured attributes, KT-PSP-25 also contains rich records of students' interaction sequences, including student IDs, question IDs, and their handwritten problem-solving processes. Because such data may expose personal information, we implemented privacy-preserving measures. Student IDs and question IDs were mapped to non-reversible digital identifiers to ensure anonymity. For the handwritten problem-solving processes, we applied 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 used in the OCR pipeline are provided in Appendix A.1

4.2 Dataset Analysis

Our dataset contains two categories of attributes: problem-level attributes and student interaction records. The problem-level attributes consist of 2,696 unique problem IDs along with their corresponding problem texts. Each problem is annotated with a ground-truth answer and solution explanation. Furthermore, all problems are tagged with one of 490 unique KCs and are categorized into five distinct difficulty levels and two question types, multiple-choice and short-answer. The student interaction attributes are composed of 1,343 unique student IDs together with each student's selected answer, answer correctness, and response duration. In addition, we collect each student's problem-solving process, which is especially important in mathematics and allows us to estimate math proficiency beyond simple correctness. Accordingly, we record problem-solving process data for all 22,289 student interactions in the dataset.

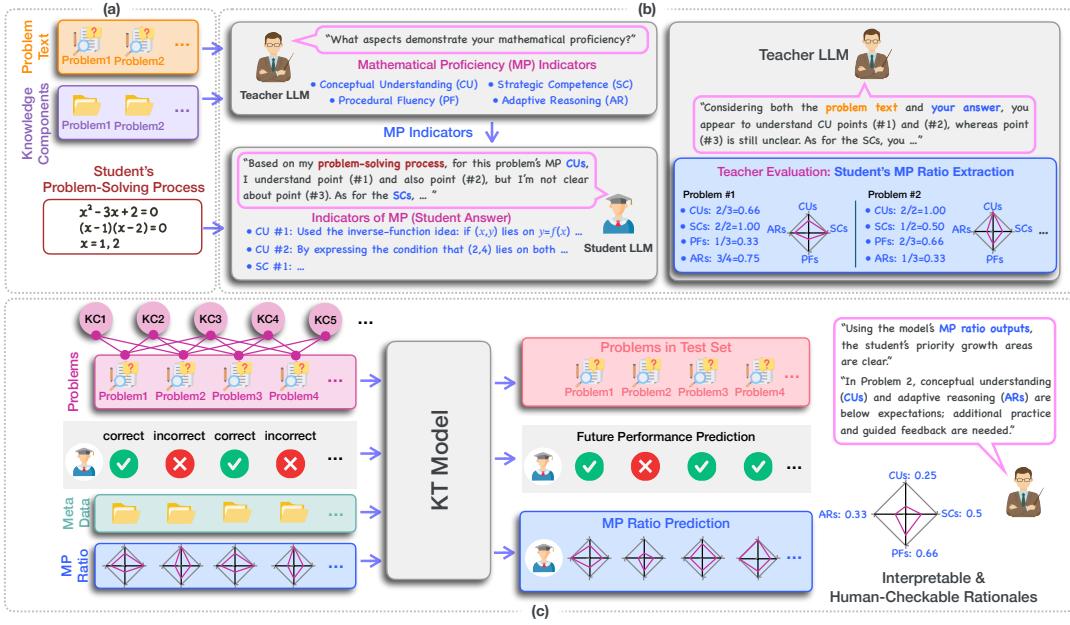


Figure 2: StatusKT Framework Overview. (a) Students’ problem-solving processes in math tasks provide rich cues about their understanding, which our framework leverages to derive mathematical proficiency (MP) indicators. (b) A teacher LLM analyzes each problem’s text and knowledge components to construct MP indicators, while 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. (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.

5 Methodology

This section presents the details of STATUSKT, illustrated in Figure 2. STATUSKT enhances knowledge tracing by explicitly modeling students’ mathematical proficiency (MP) through interpretable signals derived from their problem-solving processes. While conventional KT models rely on binary correctness, our approach incorporates structured proficiency evidence generated from the underlying problem-solving process. 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 of its difficult nature to reliably assess 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 2(a)). These traces contain interpretable signals

of students’ reasoning capabilities. Our pipeline is inspired by the structured evaluation protocol in El-Shara et al. (2025), which assesses students through fixed MP indicators answered directly by students and evaluated by teachers. In contrast, our framework cannot rely on fixed indicators or direct student interviews, so we employ a three stage Teacher-Student-Teacher LLM pipeline (Figure 2(b)) to extract MP from the raw problem-solving processes:

1. Indicator Extraction(Teacher LLM). Given the problem text, the first Teacher LLM generates a set of MP indicators, each phrased as a question targeting a specific proficiency dimension. These indicators act as a rubric specifying the reasoning elements, and metadata for the problem guiding the subsequent analysis.

2. Response Generation (Student LLM).

Given the MP indicators, the Student LLM generates responses based on the problem-solving process. This step converts the implicit reasoning in the problem-solving process into explicit evidence corresponding to each proficiency dimension.

3. Proficiency Assessment (Teacher LLM).

nally, the teacher LLM evaluates whether each response satisfies its corresponding indicator. Based on these evaluations, we compute MP Ratio for each dimension as the proportion of satisfied indicators relative to the total number of generated indicators for that dimension.

In all three stages, we utilized GPT-5 (OpenAI, 2025) model. The specific prompts designed for each step are provided in Appendix A.2.

5.2 Integrating MP Ratio for Interpretable Knowledge Tracing

The computed MP ratio serves as an interpretable intermediate representation that complements traditional KT signals. As depicted in Figure 2(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 predictive performance and interpretability beyond what correctness alone provides.

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

$$L = \text{BCE}(r_{\text{gt}}, r_{\text{pred}}) + \alpha \sum_{i \in P} \text{MSE}(m_i^{\text{gt}}, m_i^{\text{pred}}), \quad (1)$$

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

6 Experiments

In this section, we conducted experiments on KT-PSP-25 to evaluate our proposed STATUSKT framework.

6.1 Experiment Setting

6.1.1 Baseline

We evaluate the performance of STATUSKT with well-known 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).

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.

6.1.2 Implementation Details

All experiments were conducted using the pyKT library (Liu et al., 2022), a PyTorch-based framework for knowledge tracing. Training is performed on a single NVIDIA RTX 3090 GPU. The dataset is split into 80% for training and validation, and 20% test sets. To obtain reliable results given the limited dataset size, we use 10% of the training data for validation and hyperparameter tuning and early stopping with a patience of 10 epochs. All the models are trained with the ADAM optimizer (Adam et al., 2014), with a batch size of 16 for training. We search learning rate from $[5 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}, 1 \times 10^{-4}]$ and dropout rate from $[0.5, 0.3, 0.1, 0.05]$. The embedding dimensions are set to 200 for RNN-based models and 256 for Transformer-based models. A fixed

Method	DKT	DKT+	DKVMN	SKVMN	SAKT	SAINT	AKT	simpleKT	stableKT	robustKT
<i>AUC</i>										
Original	0.6171	0.6197	0.6095	0.5832	0.5867	0.6238	0.6555	0.6591	0.6706	0.6403
STATUSKT	0.6206	0.6214	0.6220	0.5981	0.5917	0.6368	0.6629	0.6629	0.6727	0.6451
<i>ACC</i>										
Original	0.7226	0.7254	0.7349	0.7397	0.7398	0.7472	0.7379	0.7393	0.7424	0.7391
STATUSKT	0.7252	0.7273	0.7410	0.7342	0.7448	0.7462	0.7423	0.7426	0.7434	0.7391

Table 2: Overall performance comparison across KT architectures with and without STATUSKT framework. Despite a few models (SKVMN, SAINT) showing small ACC decreases, the overall trend shows that incorporating students’ problem-solving processes leads to more representations and more accurate next-step performance prediction.

random seed (42) is used for reproducibility.

6.1.3 Evaluation Metrics

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

6.2 Overall Performance

To validate the effectiveness of our proposed STATUSKT framework, we applied it to KT-PSP-25 across a wide range of existing KT architectures. The experimental results are presented in Table 2. The results demonstrate that incorporating students’ problem-solving processes through STATUSKT consistently improves model performance, yielding higher AUC and ACC in most cases compared to the original versions.

Classic recurrent models such as DKT and DKVMN show clear performance gains, indicating that additional information derived from students’ solution processes provides meaningful signals for estimating their mastery states. Attention-based models (SAKT, AKT) display similar improvements, suggesting that enriched interaction representations benefit architectures with stronger capacity for dependency modeling. Lightweight models including simpleKT, stableKT, and robustKT also exhibit positive gains despite their minimal design, demonstrating the broad applicability of STATUSKT across heterogeneous model families.

Although a few models show slight decreases in AUC while increasing ACC (e.g., SKVMN, robustKT), these cases reflect shifts in ranking sensitivity rather than degradation in prediction quality.

Overall, the consistent upward trend across architectures confirms that leveraging students’ problem-solving processes leads to more informative representations and more accurate next-step performance prediction.

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.

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).
- 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.
- 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.
- Rui Luo, Fei Liu, Wenhao Liang, Yuhong Zhang, Chenyang Bu, and Xuegang Hu. 2022. Dagkt: Difficulty and attempts boosted graph-based knowledge tracing. In *International Conference on Neural Information Processing*, pages 255–266. Springer.
- 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.
- Napol Rachatasumrit and Kenneth R Koedinger. 2021. Toward improving student model estimates through assistance scores in principle and in practice. *International Educational Data Mining Society*.
- Alexis Ross, Megha Srivastava, Jeremiah Blanchard, and Jacob Andreas. 2025. Modeling student learning with 3.8 million program traces. *arXiv preprint arXiv:2510.05056*.
- 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 Tschigale, 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*.
- Bihai Xu, Zhenya Huang, Jiayu Liu, Shuanghong Shen, Qi Liu, Enhong Chen, Jinze Wu, and Shijin Wang. 2023. Learning behavior-oriented knowledge tracing. In *Proceedings of the 29th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2789–2800.
- 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 Prompts

A.1 OCR

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

- Figure 3: A prompt for GPT-based OCR to convert students' handwritten problem-solving process into text.
- Figure 4: A prompt for refining GPT-based OCR outputs to improve transcription quality.

A.2 Mathematical Proficiency Ratio Extraction

The prompts used in our STATUSKT for extracting Mathematical proficiency(MP) ratio are as follows:

- Figure 5: A prompt to extract MP indicators from problem statement and curricular unit name.
- Figure 6: A prompt for student LLM, which simulates how a student would respond to each of the generated indicators.
- Figure 7: 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 3: 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 4: 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 5: 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):
 $y^2 + 3y + 2 = 0$
 $(r+1)(r+2)=0$

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 6: 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  $y=\frac{2x-3}{2x+5}$ , 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  $y = \frac{(2x - 3)}{(2x + 5)}$  whose  $x$ - and  $y$ -values are both integers."}, {"CU2": "Identify the domain restriction  $2x + 5 \neq 0$ , 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  $y = \frac{(2x - 3)}{(2x + 5)}$  into  $y - 1 = \frac{[(2x - 3) - (2x + 5)]}{(2x + 5)} = -\frac{8}{(2x + 5)}$ ."}, {"SC2": "Since  $y$  must be an integer,  $-\frac{8}{(2x + 5)}$  must be an integer; thus reinterpret this as the divisibility condition  $2x + 5 \mid 8$ ."}, {"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:  $2x + 5 = 1 \Rightarrow x = -2$ ;  $2x + 5 = -1 \Rightarrow x = -3$ ."}, {"PF4": "For each  $x$ , compute  $y$  using  $y = 1 - \frac{8}{(2x + 5)}$ : for  $x = -2 \Rightarrow y = -7$ ; for  $x = -3 \Rightarrow y = 9$ ."}, {"PF5": "Verify that the points  $(-2, -7)$  and  $(-3, 9)$  satisfy the original equation  $y = \frac{2x - 3}{2x + 5}$ ."}, {"AR2": "Provide reasoning that only  $\pm 1$  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  $x = -\frac{5y + 3}{2(y - 1)}$ . Let  $d = y - 1$ . Then  $x = -\frac{5}{2} - \frac{4}{d}$ , and  $x$  is an integer only when  $d = \pm 8$ , 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  $y = \frac{(2x - 3)}{(2x + 5)}$  as  $y = -\frac{8}{(2x + 5)} + 1$  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  $2x + 5 \neq 0$  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  $y = \frac{(2x - 3)}{(2x + 5)}$  as  $y = -\frac{8}{(2x + 5)} + 1$ ."}, {"PF1": "Although intermediate algebra steps were omitted, the final expression  $y - 1 = -\frac{8}{(2x + 5)}$  was obtained."}, {"SC2": "They interpreted the requirement that  $-\frac{8}{(2x + 5)}$  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:  $2x + 5 = 1 \Rightarrow x = -2$ , and  $2x + 5 = -1 \Rightarrow x = -3$ . The remaining cases were left incomplete or not shown."}, {"PF4": "For  $x = -2$  and  $x = -3$ , the  $y$ -values were left blank; but likely  $x = -2 \Rightarrow y = -7$ , and  $x = -3 \Rightarrow y = 9$ ."}, {"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 7: Prompt used for evaluates the appropriateness of each generated response for its corresponding indicator in STATUSKT. Prompt inputs are **boldfaced**.