Leveraging LLMs for Fine-grained Knowledge Component Extraction in Educational AI

Anonymous ACL submission

Abstract

AI tutoring systems require fine-grained skill representations to diagnose student knowledge gaps and deliver targeted instruction. However, existing automated Knowledge Component (KC) extraction methods lack principled strategies for controlling granularity and do not systematically capture cognitive operations required for problem-solving. We present CogKC, a multi-stage LLM-based method that generates think-aloud protocols to externalize problem-solving reasoning, then extract hierarchical KCs with explicit cognitive operations (e.g., recall, calculate, infer). Our approach produces finer-grained representations than expert annotations (72.09% more specific) while maintaining interpretability. Evaluation through personalized question generation and tutoring simulation demonstrates improved question quality (68.2% win rate) and tutoring efficiency (14% gain) compared to baseline methods.

1 Introduction

The advent of Large Language Models (LLMs) has transformed AI tutoring, enabling personalized education that adapts to individual students' knowledge gaps rather than following fixed curricula (Maurya et al., 2025; Chu et al., 2025). However, effective personalization requires a fundamental design decision: at what granularity should student knowledge be diagnosed? (Aleven et al., 2023) A coarse label like "solving word problems" cannot distinguish *what* knowledge is required (inverse relationships, foundational concepts) or *how* it must be applied (recalling, calculating, reasoning). Fine-grained representations capturing both dimensions enable precise gap diagnosis, yet expert annotation at this granularity is prohibitively expensive.

Researchers formalize such knowledge units as Knowledge Components (KCs) (Aleven and Koedinger, 2013)—discrete units that ground student modeling and adaptive learning systems. Traditionally, domain experts manually annotate required KCs for each problem. Recent automated approaches using LLMs provide scalable alternatives to expert annotation (Moore et al., 2024; Duan et al., 2025).

However, current automated approaches face critical limitations for targeted instruction. While finer-grained KCs improve personalization (Huang et al., 2021; Pardos et al., 2007; Duan et al., 2025), current approaches produce inconsistent representations that mix broad concepts with atomic procedures, without explicit criteria for controlling granularity (Moore et al., 2024). Furthermore, existing approaches (Moon et al., 2025; Duan et al., 2025) primarily focus on *what* knowledge is required, and do not explicitly model *how* it must be applied.

Compounding these challenges, validating fine-grained KC extraction poses a fundamental problem. Statistical methods such as Knowledge Tracing (Corbett and Anderson, 1994; Scarlatos et al., 2025) and Learning Factor Analysis (Cen et al., 2006) require large-scale student response data. Also, comparison against expert labels cannot assess granularities finer than experts themselves provide (Moore et al., 2024; Moon et al., 2025).

We address these limitations through *CogKC* (Cognitive Knowledge Components), a multi-stage LLM-based KC extraction method (Figure 1). Our key insight is to leverage LLM-generated think-aloud protocols that verbalize reasoning step-by-step as an intermediate representation that makes cognitive operations explicit. Because each verbalized step naturally corresponds to a specific cognitive action (e.g., recalling a formula, executing a calculation), we can systematically map these steps to cognitive operations using the TIMSS framework (Mullis et al., 2021). Each extracted KC specifies both the cognitive operation and knowledge content, grounded in relevant think-aloud excerpts. Our method further enables granularity con-

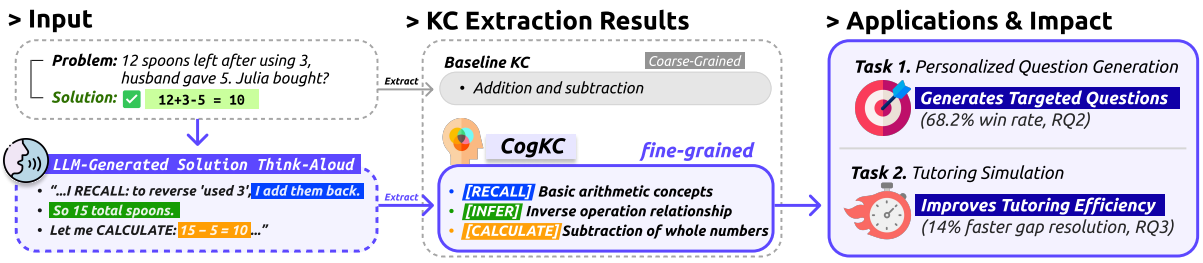

Figure 1: Overview of CogKC. Think-aloud protocols enable fine-grained KC extraction with cognitive operations, improving question generation and tutoring efficiency.

trol through recursive decomposition of these KCs.

To validate fine-grained KCs beyond the limitations of statistical methods or expert-label comparison, we propose a task-based evaluation framework using personalized question generation and tutoring conversation simulation. Through empirical evaluation, we demonstrate that 72.09% of our extracted KCs are more specific than corresponding expert annotations. This finer granularity leads to improved downstream performance, achieving up to 68.2% win rate in question generation and 14% efficiency gain in tutoring. The contributions of our paper are as follows:

- We leverage LLM-generated think-aloud reasoning to enable fine-grained KC extraction with cognitive operations, without expert annotation.
- We propose a task-based evaluation framework for fine-grained KC quality using question generation and tutoring simulation, circumventing the need for large-scale student data or fine-grained expert labels.
- We empirically demonstrate that finer-grained KCs improve downstream pedagogical outcomes.

2 Related Work

2.1 Knowledge Components in Educational Systems

Knowledge Components (KCs) (Aleven and Koedinger, 2013) represent discrete units of knowledge or skills required to solve problems, also called knowledge concepts (Chau et al., 2021) or learning objects (Liu et al., 2024). A key focus in KC research is the mapping between skills and specific problems (Aleven and Koedinger, 2013; Barnes, 2005).

This approach is compatible with traditional student modeling methods such as Cognitive Diagnosis Models (CDM) (Templin and Henson, 2006) and Knowledge Tracing (KT) (Corbett and Anderson, 1994; Ozyurt et al., 2025; Scarlatos et al., 2025). KC representations have been widely adopted in Intelligent Tutoring Systems (Aleven et al., 2025; Luburić et al., 2025; Bernd and Chounta, 2024; Smith et al., 2024; Huang et al., 2021) and educational datasets (Yu et al., 2023, 2021; Stamper et al., 2011; Stamper and Pardos, 2016), providing foundational infrastructure for personalized learning (Xiong et al., 2024).

However, while multiple studies emphasize that finer-grained KCs improve personalization (Huang et al., 2021; Pardos et al., 2007; Duan et al., 2025; Liu et al., 2024; Yu et al., 2023; Pardos et al., 2006), the term "fine-grained" lacks operational definition. Existing methods provide no principled strategy for controlling granularity or decomposing coarse-grained skills using explicit criteria. While Cognitive Task Analysis (Clark et al., 2008)–a structured method for decomposing complex tasks into constituent skills–could enable such principled decomposition, Moore et al. (2024) note it remains expensive and labor-intensive, making "fine-grained" aspirational rather than actionable without automated methods.

2.2 Automated KC Extraction and Validation

Recent advances in LLMs have enabled automatic KC extraction using prompting or fine-tuning (Moore et al., 2024; Duan et al., 2025; Moon et al., 2025; Wei et al., 2025; Ozyurt et al., 2024a; Chau et al., 2021; Thaker et al., 2019), as well as assigning KCs from established taxonomies to problems (Kwak and Pardos, 2024; Liu et al., 2024; Shi et al., 2024; Moon et al., 2024; Ozyurt et al., 2024b; Sampaio de Alencar et al., 2025). However, current automatic KC extraction methods do not explicitly

model cognitive dimensions. While manual annotation efforts (Yu et al., 2023) have explored cognitive labeling based on Bloom's taxonomy (Krathwohl, 2002), automatic approaches (Moon et al., 2025; Duan et al., 2025) do not structure their extraction to capture the cognitive operations required for problem-solving, failing to distinguish whether students struggle with recalling concepts, executing procedures, or applying higher-order reasoning.

To validate these generated labels, studies compare them against expert-generated labels or evaluate student modeling performance through Knowledge Tracing (Scarlatos et al., 2025) and Learning Factor Analysis (Cen et al., 2006). However, existing validation approaches face limitations for assessing fine-grained KCs. Statistical methods require large-scale data, while comparison against expert labels cannot assess granularities finer than experts themselves provide (Moon et al., 2025). Given the emergence of AI tutors, evaluating whether KC representations enable effective personalized instruction remains critical. As Srivastava and Goodman (2021) note, adaptive systems maximize effectiveness by generating learner-appropriate materials. Personalized question generation (Kulshreshtha et al., 2022; Hang et al., 2024; Lin et al., 2024) represents a particularly diagnostic task requiring precise KC-level understanding, yet no prior work validates KC quality through such downstream pedagogical tasks.

3 Knowledge Component (KC) Extraction

We now present our four-stage extraction pipeline that produces hierarchical multi-faceted KC representations. Our method takes as input a problem statement and its solution. While domain-agnostic, we illustrate it using mathematics, where the TIMSS cognitive framework (Mullis et al., 2021) provides expert-designed action taxonomies (e.g., knowing, applying, reasoning). Figure 2 illustrates the complete pipeline, which processes problem-solution pairs through: (1) think-aloud generation, (2) reasoning step extraction, (3) preliminary KC extraction, and (4) hierarchical decomposition.

3.1 KC Representation

Each KC captures four dimensions (Figure 2): (1) cognitive action describing the mental operation required, (2) content element describing the domain concept, (3) learning outcome specifying what students can demonstrate, and (4) relevant think-aloud providing concrete reasoning evidence. Cognitive actions are drawn from TIMSS (Mullis et al., 2021). KCs form a hierarchical structure where Layer 0 represents high-level concepts and Layer 1+ decomposes them into more specific sub-components.

3.2 KC Extraction Pipeline

Our LLM-based extraction pipeline uses think-aloud as an intermediate representation for hierarchical KC decomposition.

Stage 1: Generate think-aloud. Given a problem and solution, an LLM generates a think-aloud that simulates expert problem-solving reasoning, applying TIMSS cognitive actions throughout (prompt in Appendix A.3), producing human-like reasoning traces (Xie et al., 2025; Yu et al., 2025). This generated think-aloud grounds subsequent KC extraction steps.

Stage 2: Extract structured reasoning steps. An LLM extracts a sequence of reasoning steps from the think-aloud text, where each step includes a description and direct quotes as evidence. Steps are grouped when they involve similar operations to reduce redundancy (prompt in Appendix A.4).

Stage 3: Extract preliminary KCs from reasoning steps. An LLM infers knowledge components relevant to each reasoning step group by identifying cognitive actions and mapping them to actions in the TIMSS framework (prompt in Appendix A.5). The LLM is instructed to generate multi-faceted KCs by describing its cognitive action, content element, learning outcome, and supporting think-aloud evidence. This step produces Layer 0 KCs, which represent high-level concepts and serve as seeds for hierarchical decomposition.

Stage 4: Decompose KCs into subcomponents. A prompted LLM then decomposes each Layer 0 KC into more specific Layer 1 KCs (prompt in Appendix A.6). Specifically, the LLM is grounded in the think-aloud, which grounds KCs in concrete reasoning steps and produces hierarchies that reflect the problem-solving procedure. While the decomposition step can be applied recursively (i.e., multiple times), we only apply it once, as initial tests revealed that deeper decomposition leads to unnecessary fragmentation of concepts.

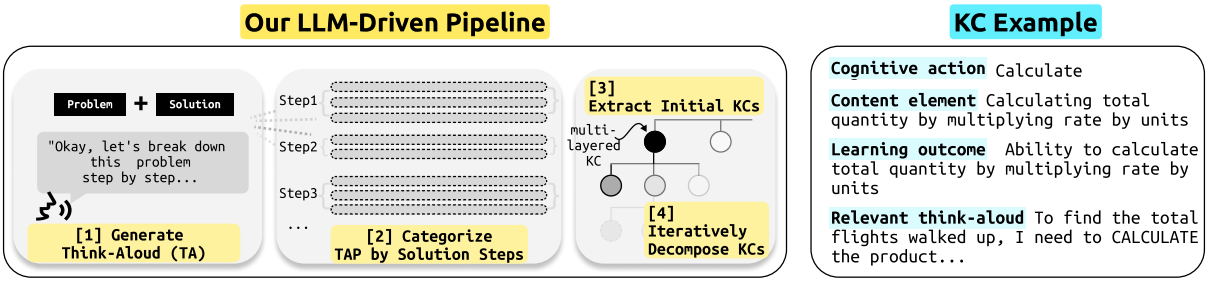

Figure 2: **CogKC Extraction Pipeline and KC Example.** (Left) The pipeline processes problems through four stages: (1) generating detailed think-aloud, (2) extracting structured reasoning steps, (3) extracting preliminary KCs from reasoning steps, and (4) decomposing KCs into hierarchical subcomponents. (Right) An example KC showing the four dimensions: cognitive action, content element, learning outcome, and relevant think-aloud.

3.3 KC Merging

We apply two-stage intra-problem merging to eliminate redundancy while preserving semantic distinctions. After Stage 3, we merge similar Layer 0 KCs based on embedding similarity and LLM-based semantic evaluation. After Stage 4, we apply hierarchical merging using FAISS (Douze et al., 2024) clustering, where an LLM determines whether to merge equivalent KCs, establish parent-child relationships, or keep distinct KCs separate. See Appendix A.7 for details.

4 Extraction Quality Validation

We address the following research question using the ATC dataset.

- **RQ1**: Can CogKC extract KCs that comprehensively cover expert-annotated knowledge components?

4.1 Experimental Settings

4.1.1 Dataset

To assess whether our method can capture KCs authored by experts, we use the Achieve the Core (ATC) dataset.[1] This dataset consists of Common Core State Standards (CCSS), a U.S. mathematics curriculum framework that offers comprehensive coverage of KCs involved in K-12 math education. Each KC or standard in the dataset includes a label (e.g., *2.MD.A.3*), a description (e.g., *Estimate lengths using units of inches...*), and a set of related problems.

4.1.2 Baseline Methods

We compare our hierarchical KC extraction approach against two baseline methods: *LLMKT* (Moon et al., 2025) and *KCQRL* (Ozyurt et al., 2024a). Both baselines produce KCs as single descriptive strings (e.g., "solving linear equations"), lacking our multi-faceted structure (cognitive action, content, learning outcome, think-aloud evidence) and hierarchical decomposition.

Baseline methods typically extract fewer, less detailed KCs. To isolate structural advantages from information quantity, we implement three variants: (i) *Orig*—original method, (ii) *Basic*—KC count matched to our method, and (iii) *Full*—both KC count and per-KC information matched or exceeded through augmented prompts eliciting detailed descriptions. This ensures observed advantages stem from a hierarchical multi-faceted structure rather than information quantity. Complete baseline specifications, prompts, and information content validation are in Appendix B.

4.1.3 Implementation Details

We apply our method (Section 3) to each problem in the ATC dataset, extracting hierarchical KCs with intra-problem merging. We evaluate two configurations: *ML0* extracts Layer 0 KCs without recursive decomposition (Stage 4 skipped), and *ML1* applies one round of recursive decomposition to produce Layer 0 and Layer 1 KCs. Both configurations can form hierarchical structures through intra-problem merging. Each problem's KCs are evaluated against the corresponding gold standard. We compare against the baseline methods and controlled variants described above.

4.1.4 Metrics

We evaluate the KC extraction performance of our method against the gold KCs in the ATC dataset by (1) retrieving semantically similar KC pairs, (2) labeling their relation, and (3) calculating precision,

[1] http://achievethecore.org/coherence-map/; data available at https://huggingface.co/datasets/allenai/achieve-the-core

recall, and F_1. Let $K_G = \{k_g\}$ be the gold KCs and $K_P = \{k_p\}$ be the predicted KCs.

1. Candidate retrieval. For each predicted KC, we concatenate its concept element c and learning outcome l for similarity comparison, excluding cognitive action and think-aloud parts that are specific to our multi-faceted representation and not present in gold standard KCs. We calculate sentence embeddings for these descriptions and the gold KCs, compute cosine similarity for all pairs $(k_g, k_p) \in K_G \times K_P$, and keep those with similarity above 0.5 as *candidate pairs*.

2. Relation classification. Each candidate pair (k_g, k_p) is passed to an LLM (`gemini-2.5-flash`, temperature=0.1) to label if: (1) k_g is *equivalent* to k_p, (2) k_g is a *higher*-level concept than k_p, (3) k_g is *lower*-level than k_p, or (4) *no relationship*. The relation is denoted $r(k_g, k_p) \in \{\text{eq}, \text{high}, \text{low}, \text{none}\}$.

3. Scoring. We count a predicted KC as encompassing a gold KC when the gold KC represents an *equivalent* or a *higher*-level concept than the predicted KC. We define precision P and recall R as follows:

$$P = |\{k_p \mid \exists\, k_g : r(k_g, k_p) \in \{\text{eq}, \text{high}\}\}| / |K_P|$$

$$R = |\{k_g \mid \exists\, k_p : r(k_g, k_p) \in \{\text{eq}, \text{high}\}\}| / |K_G|$$

We validated this metric by collecting human annotations for 150 gold-predicted KC pairs, leading to a Krippendorff's alpha (Krippendorff, 2004) of 0.448 (*moderate agreement*) with the majority vote of human annotators. Details in Appendix A.8.

Method	$\lvert K_P \rvert$	P	R	F_1
CogKC-ML0	2260	0.681	0.912	0.780
CogKC-ML1	5443	0.678	0.965	0.796
LLMKT-ORIG	753	**0.915**	0.924	**0.920**
LLMKT-BASIC	5470	0.734	**0.984**	0.841
LLMKT-FULL	5477	0.739	0.972	0.839
KCQRL-ORIG	416	0.714	0.786	0.748
KCQRL-BASIC	3999[2]	0.624	0.959	0.756
KCQRL-FULL	5477	0.783	0.959	0.817

Table 1: KC extraction performance on ATC dataset. $|K_P|$: KC count; P: precision; R: recall.

[2]KCQRL-BASIC extracted 5477 nodes total, but $|K_P|$ was calculated after removing duplicates.

Method	Relationship Label		
	eq	high	low
CogKC-ML0	4.90%	67.38%	27.72%
CogKC-ML1	4.67%	72.09%	23.24%
LLMKT-ORIG	9.46%	62.01%	28.54%
LLMKT-BASIC	7.08%	62.40%	30.52%
LLMKT-FULL	7.36%	60.81%	31.83%
KCQRL-ORIG	9.18%	51.15%	39.67%
KCQRL-BASIC	7.60%	58.90%	33.50%
KCQRL-FULL	5.76%	58.82%	35.42%

Table 2: Distribution of relation labels for gold-predicted KC pairs. eq: equivalent; high: gold KC broader; low: gold KC narrower.

4.2 Results & Analysis

Tables 1 and 2 compare performance against 317 expert-annotated KCs. CogKC-ML1 maintains high recall (0.965), indicating comprehensive coverage of expert-annotated knowledge. The key finding is in Table 2: 72.09% of CogKC's matched pairs have gold KCs labeled as broader (*high*), compared to 60-62% for LLMKT and 51-59% for KCQRL, confirming CogKC extracts finer-grained KCs than expert standards. Granularity increases with hierarchical decomposition from ML0 (67.38%) to ML1 (72.09%). See Appendix A.9 for qualitative examples. This finer granularity naturally yields lower precision (0.678), as atomic sub-skills do not directly match broader expert labels. Since expert-label comparison cannot validate granularities finer than experts provide, we evaluate through downstream pedagogical tasks in Section 5.

5 Downstream Personalization Validation

Having established that our method produces finer-grained KCs than expert annotations and baseline methods (Section 4), we now investigate whether this translates to improved downstream task performance. We propose two evaluation tasks as our task-based evaluation framework (Figure 3), validating KC quality through downstream pedagogical outcomes—reasoning that high-quality KCs should yield better personalization.

Specifically, we evaluate through personalized question generation (Kulshreshtha et al., 2022; Hang et al., 2024; Lin et al., 2024), which requires precise KC-level gap identification where granu-

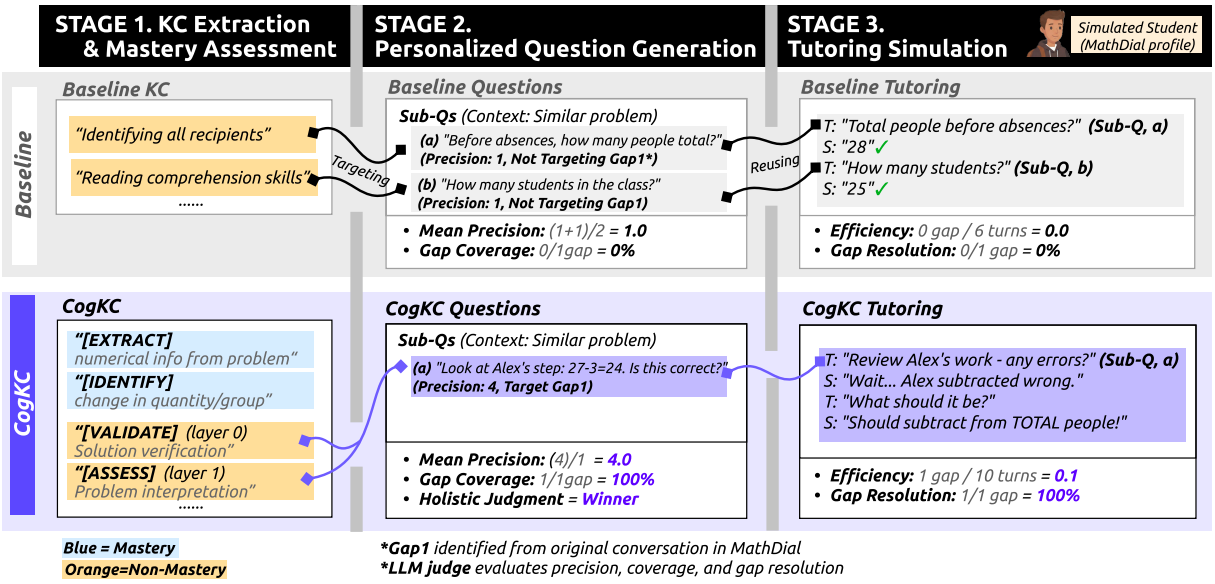

Figure 3: Downstream evaluation workflow. KC mastery assessment informs question generation (RQ2), which drives tutoring simulation (RQ3). CogKC's precise KCs enable accurate gap targeting compared to baseline.

larity directly impacts quality. We address two research questions using the MathDial dialogue tutoring dataset (Macina et al., 2023):

- **RQ2**: Does finer-grained KC extraction improve personalized question generation?
- **RQ3**: Do these improvements translate to more effective tutoring outcomes?

5.1 RQ2: Personalized Question Generation

5.1.1 Task Description and Setup

We evaluate whether finer-grained KC extraction improves personalized follow-up question generation. The task requires generating targeted questions that address specific knowledge gaps identified from student-tutor conversations.

Dataset. We use the MATHDIAL dataset (Macina et al., 2023), a dialogue tutoring corpus comprising 2,861 one-to-one teacher-student conversations grounded in multi-step math reasoning problems from GSM8k (Cobbe et al., 2021). The dataset was collected using a semi-synthetic approach pairing human teachers with LLM-simulated students, with annotations including math word problems and solutions.

Question Generation Pipeline. All KC-based methods follow a three-stage pipeline:

1. **KC Extraction**: Extract problem-solving skills from mathematical problems
2. **KC Mastery Assessment**: Classify each KC as *mastery* (clear understanding), *non-mastery* (confusion/errors), or *insufficient info* (insufficient evidence)
3. **Question Generation**: Create one main problem with sub-questions targeting *non-mastery* KCs

For CogKC, when student and teacher solution methods differ, we extract KCs for both approaches and merge them to capture diverse problem-solving strategies. See Appendix C.1 for detailed classification guidelines and Appendix C for complete pipeline prompts.

Compared Methods. We compare CogKC against two baseline KC extraction methods (LLMKT, KCQRL) using information-controlled variants (Basic, Full; see §4.1.2 for details) and a Vanilla baseline that directly analyzes conversations without intermediate KC decomposition. To assess the contribution of hierarchical structure and cognitive operations, we test CogKC ablations: ML1 (full method), ML1-CO (without cognitive operations), and ML0 (without hierarchy). All methods use `gemini-2.5-flash` with temperature=0.1.

Evaluation Methodology. We employ pairwise LLM-as-judge evaluation on all 2,861 conversations (see Appendix C.4 for validation). After standardized gap identification, we generate questions

with each method and conduct pairwise comparisons. To mitigate positional bias, we evaluate each pair in both orderings. We measure three metrics: (i) Gap Targeting Precision (1-4 scale), (ii) Gap Coverage (percentage of gaps addressed), and (iii) Overall—since precision and coverage can trade off, this holistic judgment balances both factors.

5.1.2 Results & Analysis

Table 3 presents pairwise comparisons for CogKC-ML1 against all baselines.

Table 3: Question Generation Performance (RQ2, N=2,861). CogKC-ML1 win rates (%) vs baselines. See Appendix Table 20 for full details.

Baseline	Overall	Precision	Coverage
LLMKT-FULL	**56.3*****	**61.7**	**84.3**
KCQRL-FULL	**60.6*****	**66.8**	**83.7**
Vanilla	**67.1*****	**68.2**	**85.5**

***$p < 0.001$ (paired t-test).

Question Targeting and Coverage Advantages. CogKC-ML1 significantly outperforms all baselines across holistic judgment, precision, and coverage metrics. CogKC's advantages over LLMKT-FULL (56.3%) and KCQRL-FULL (60.6%) demonstrate the benefits of hierarchical multi-faceted structure with explicit cognitive operations over flat single-faceted representations. The stronger advantage against Vanilla (67.1%) further confirms the overall value of KC-based methods. Notably, coverage advantages (>83%) exceed precision advantages (61-68%), suggesting that while hierarchical KCs improve targeting quality, their greatest strength lies in enabling comprehensive gap coverage.

Ablation Analysis. Appendix Table 20 reveals both design components are essential. Removing cognitive operations (ML1-CO) drops performance to baseline levels (50.0%), while removing hierarchy (ML0) underperforms LLMKT-FULL (44.1% vs 53.9%). Only the full ML1 configuration—combining hierarchical structure with operational specifications—achieves consistent advantages.

Information Volume Controls. CogKC-ML1's advantages persist across information-controlled variants. Win rates against BASIC (LLMKT: 59.2%, KCQRL: 64.5%) and FULL (56.3%, 60.6%) confirm hierarchical structure drives improvements.

5.2 RQ3: Tutoring Effectiveness

5.2.1 Task Description and Setup

We evaluate whether better-targeted questions (RQ2) translate to more effective tutoring outcomes. Using questions generated in RQ2, we simulate full tutoring conversations with pairwise comparisons using the same evaluation setup.

Dataset. We use a subset of 1,399 conversations from the MathDial dataset (599 from test set, 800 from training set) for tutoring simulation.[3] We compare CogKC-ML1 against LLMKT-FULL baseline using the same method configurations as RQ2. All simulations use `gemini-2.5-flash-lite` (for computational efficiency) with temperature=0.0 for tutor, student, and evaluator roles.

Tutoring Simulation Pipeline. Each simulation begins with the generated question set from RQ2. We use LLMs to simulate tutors and students. The tutor works through questions while providing guidance and scaffolding, while the student responds based on their knowledge profile and confusion patterns from original MathDial conversations. An LLM-as-Judge evaluator assesses gap resolution by analyzing the conversation trajectory. Conversations run until completion or reach the maximum turn limit (25 turns). To ensure fairness, we use LLM-identified knowledge gaps rather than method-specific KC assessments, eliminating bias from differing KC representations (see Appendix D.1 for details).

Evaluation Methodology. We measure tutoring effectiveness through three complementary metrics: (i) Efficiency—gaps resolved per turn (higher is better), (ii) Gap Resolution Ratio—percentage of identified gaps resolved, and (iii) Completion Rate—percentage completing before turn limit, regardless of gap resolution. For comparative analysis, we report win rates: percentage of conversations where one method achieves higher efficiency or resolves more gaps. See Appendix D.3 for detailed metric formulas and calculation examples.

5.2.2 Results & Analysis

Table 4 presents the effective metrics for pairwise comparisons between CogKC-ML1 and LLMKT-

[3] Sample size selection were determined through convergence analysis and diminishing returns testing. See Appendix D.2 for complete statistical validation.

FULL across 1,399 simulated tutoring conversations.

Table 4: Tutoring Effectiveness (RQ3, N=1,399). CogKC-ML1 vs LLMKT-FULL. ***$p < 0.001$, *$p < 0.05$ (paired t-test); n.s.=not sig.

Metric	CogKC	LLMKT	Sig.
Absolute Performance			
Efficiency (gaps/turn)	**0.122**	0.107	***
Gap Resolution Ratio	**66.4%**	63.6%	*
Completion Rate	76.8%	77.1%	n.s.
Win Rates			
Efficiency Win	**47.4%**	39.5%	+7.9pp
Gap Resolution Win	**28.4%**	23.7%	+4.7pp

***$p < 0.001$, *$p < 0.05$ (paired t-test); n.s.=not sig.

Efficiency and Gap Resolution Advantages. CogKC-ML1 significantly outperforms LLMKT-FULL on both efficiency (0.122 vs. 0.107 gaps/turn, 14% improvement) and gap resolution metrics. Consistent win rate advantages (efficiency: +7.9pp; gap resolution: +4.7pp) demonstrate CogKC-ML1's superiority in head-to-head comparisons. While both methods achieve similar completion rates, CogKC-ML1 resolves more knowledge gaps within the same conversation length. Effect sizes are small (Cohen's $d = 0.103$) but meaningful—differences arise solely from KC representation quality under identical pedagogical approaches. See Appendix D.2 for full statistical details.

5.3 Chain of Evidence Validation

RQ2 and RQ3 evaluate question quality and tutoring effectiveness respectively. If fine-grained KCs consistently improve both stages within the same cases, this provides stronger evidence for their effectiveness. To confirm these advantages are connected rather than independent phenomena, we validate whether question quality improvements (RQ2) predict tutoring effectiveness (RQ3).

Statistical analysis across 1,399 conversations shows RQ2 winners significantly predict RQ3 winners: chi-square tests confirm associations across all criterion combinations (three RQ2 criteria × two RQ3 criteria, all $p < 0.05$). Figure 4 shows relative likelihood ratios ranging from 1.12x to 1.66x, with Coverage → Gap Resolution demonstrating the strongest relationship (1.66x, $p < 0.001$)—when CogKC wins on gap coverage in RQ2, it is 66% more likely to win on gap resolution in RQ3. This establishes the empirical link from better-targeted questions to more efficient tutoring outcomes. See Appendix E for details.

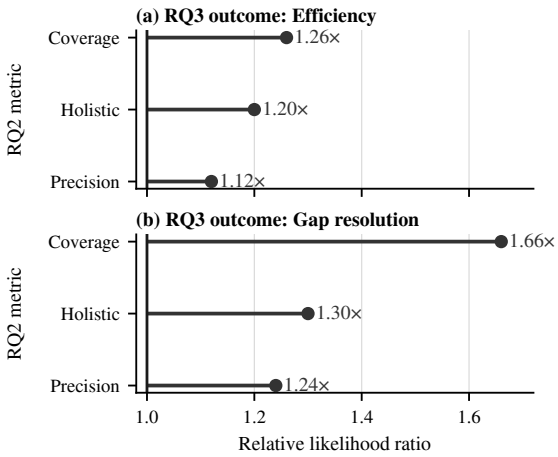

Figure 4: RQ2 question quality predicts RQ3 tutoring outcomes (N=1,399). Relative likelihood ratio measures how much more likely CogKC wins RQ3 given it won RQ2; values >1 indicate positive relationship. Coverage → Gap Resolution shows strongest effect (1.66x). All $p < 0.05$.

6 Conclusion

We introduce a hierarchical method for knowledge component extraction that bridges the gap between automated analysis and pedagogical utility. By using think-aloud protocols as intermediate representations and grounding extraction in cognitive frameworks, we enable systematic decomposition of mathematical knowledge into fine-grained, operationalizable components. Our evaluations demonstrate that extraction quality directly influences downstream pedagogical outcomes—from question targeting to tutoring efficiency—establishing a causal link between KC granularity and instructional effectiveness. This work advances educational AI toward systems that can generate targeted interventions at scale, moving beyond performance prediction to enable personalized content creation that adapts to individual learning needs.

7 Limitations

First, our approach has been validated only on K-12 mathematics tutoring using the TIMSS framework, and validation on other educational domains (e.g., science, language learning, programming) is needed to assess the generalizability of our

multi-faceted extraction approach and hierarchical layer semantics. Second, the multi-stage extraction process with exhausting prompt lengths (incorporating think-aloud protocols, problem context, and multi-layer specifications) incurs significantly higher computational costs and token usage compared to single-pass extraction methods, which may limit scalability for very large-scale applications. Third, our tutoring evaluation (RQ3) relies on simulated conversations, and the MathDial dataset itself was constructed from simulated student-human tutor interactions rather than authentic classroom dialogues, necessitating validation with real human-human tutoring sessions to confirm ecological validity.

Ethical Considerations

All human evaluation studies were conducted with institutional review board (IRB) approval; all participants provided informed consent prior to participation. For KC metric validation, we recruited 15 mathematics experts through Prolific at £11 for approximately 30 minutes (£22 per hour). For LLM-as-judge validation, we recruited 6 expert mathematics educators at 50,000 KRW (approximately 34.66 USD as of January 2026) per hour. Both rates meet or exceed fair compensation guidelines for our region. We use publicly available datasets (ATC under ODC-By 1.0, MathDial under CC BY-SA 4.0) and baseline implementations under the license terms provided by the authors in their GitHub repositories. Our use of these artifacts is consistent with their intended research purposes. We acknowledge using Claude for writing and coding assistance, and Gemini Deep Research for literature search.

References

Vincent Aleven, Conrad Borchers, Yun Huang, Tomohiro Nagashima, Bruce McLaren, Paulo Carvalho, Octav Popescu, Jonathan Sewall, and Kenneth Koedinger. 2025. An integrated platform for studying learning with intelligent tutoring systems: Ctat+ tutorshop. *arXiv preprint arXiv:2502.10395*.

Vincent Aleven and Kenneth R Koedinger. 2013. Knowledge component (kc) approaches to learner modeling. *Design recommendations for intelligent tutoring systems*, 1:165–182.

Vincent Aleven, Jonathan Rowe, Yun Huang, and Antonija Mitrovic. 2023. Domain modeling for aied systems with connections to modeling student knowledge: A review. *Handbook of Artificial Intelligence in Education*, pages 127–169.

Tiffany Barnes. 2005. The q-matrix method: Mining student response data for knowledge. In *American association for artificial intelligence 2005 educational data mining workshop*, pages 1–8. AAAI Press, Pittsburgh, PA, USA.

Nadine Schulze Bernd and Irene-Angelica Chounta. 2024. Beyond the grey area: Exploring the effectiveness of scaffolding as a learning measure. In *Artificial Intelligence in Education: 25th International Conference, AIED 2024, Recife, Brazil, July 8–12, 2024, Proceedings, Part I*, pages 365–378.

Hao Cen, Kenneth Koedinger, and Brian Junker. 2006. Learning factors analysis–a general method for cognitive model evaluation and improvement. In *International conference on intelligent tutoring systems*, pages 164–175. Springer.

Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2021. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*, 31(4):820–846.

Zhendong Chu, Shen Wang, Jian Xie, Tinghui Zhu, Yibo Yan, Jinheng Ye, Aoxiao Zhong, Xuming Hu, Jing Liang, Philip S Yu, and 1 others. 2025. Llm agents for education: Advances and applications. *arXiv preprint arXiv:2503.11733*.

Richard E Clark, David F Feldon, Jeroen JG Van Merrienboer, Kenneth A Yates, and Sean Early. 2008. Cognitive task analysis. In *Handbook of research on educational communications and technology*, pages 577–593. Routledge.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Zhangqi Duan, Nigel Fernandez, Arun Balajiee Lekshmi Narayanan, Mohammad Hassany, Rafaella Sampaio de Alencar, Peter Brusilovsky, Bita Akram, and Andrew Lan. 2025. Automated knowledge component generation and knowledge tracing for coding problems. *arXiv preprint arXiv:2502.18632*.

Ching Nam Hang, Chee Wei Tan, and Pei-Duo Yu. 2024. Mcqgen: A large language model-driven mcq generator for personalized learning. *IEEE Access*, 12:102261–102273.

Yun Huang, Nikki G Lobczowski, J Elizabeth Richey, Elizabeth A McLaughlin, Michael W Asher, Judith M Harackiewicz, Vincent Aleven, and Kenneth R Koedinger. 2021. A general multi-method approach to data-driven redesign of tutoring systems. In *LAK21: 11th International Learning Analytics and Knowledge Conference*, pages 161–172.

Matthew A Kraft. 2020. Interpreting effect sizes of education interventions. *Educational researcher*, 49(4):241–253.

David R Krathwohl. 2002. A revision of bloom's taxonomy: An overview. *Theory into practice*, 41(4):212–218.

Klaus Krippendorff. 2004. Reliability in content analysis: Some common misconceptions and recommendations. *Human communication research*, 30(3):411–433.

Devang Kulshreshtha, Muhammad Shayan, Robert Belfer, Siva Reddy, Iulian Vlad Serban, and Ekaterina Kochmar. 2022. Few-shot question generation for personalized feedback in intelligent tutoring systems. *arXiv preprint arXiv:2206.04187*.

Yerin Kwak and Zachary A Pardos. 2024. Bridging large language model disparities: Skill tagging of multilingual educational content. *British Journal of Educational Technology*, 55(5):2039–2057.

Zihao Lin, Zichao Wang, Yuanting Pan, Varun Manjunatha, Ryan Rossi, Angela Lau, Lifu Huang, and Tong Sun. 2024. Persona-sq: A personalized suggested question generation framework for real-world documents. *arXiv preprint arXiv:2412.12445*.

Naiming Liu, Shashank Sonkar, Debshila Basu Mallick, Richard Baraniuk, and Zhongzhou Chen. 2024. Atomic learning objectives labeling: A high-resolution approach for physics education. *arXiv [cs.CY]*.

Nikola Luburić, Luka Dorić, Jelena Slivka, Dragan Vidaković, Katarina-Glorija Grujić, Aleksandar Kovačević, and Simona Prokić. 2025. An intelligent tutoring system to support code maintainability skill development. *IEEE Transactions on Learning Technologies*.

Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. Mathdial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. *arXiv preprint arXiv:2305.14536*.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251.

Hyeongdon Moon, Richard Davis, Seyed Parsa Neshaei, and Pierre Dillenbourg. 2024. Using large multimodal models to extract knowledge components for knowledge tracing from multimedia question information. *arXiv preprint arXiv:2409.20167*.

Hyeongdon Moon, Richard Davis, Seyed Parsa Neshaei, and Pierre Dillenbourg. 2025. Using large multimodal models to extract knowledge components for knowledge tracing from multimedia question information. *arXiv preprint arXiv:2507.xxxxx*.

Steven Moore, Robin Schmucker, Tom Mitchell, and John Stamper. 2024. Automated generation and tagging of knowledge components from multiple-choice questions. In *Proceedings of the eleventh ACM conference on learning@ scale*, pages 122–133.

Ina VS Mullis, Michael O Martin, and Matthias von Davier. 2021. Timss 2023 assessment frameworks. *International Association for the Evaluation of Educational Achievement*.

Yilmazcan Ozyurt, Tunaberk Almaci, Stefan Feuerriegel, and Mrinmaya Sachan. 2025. Personalized exercise recommendation with semantically-grounded knowledge tracing. *arXiv preprint arXiv:2507.11060*.

Yilmazcan Ozyurt, Stefan Feuerriegel, and Mrinmaya Sachan. 2024a. Automated knowledge concept annotation and question representation learning for knowledge tracing. *arXiv preprint arXiv:2410.xxxxx*.

Yilmazcan Ozyurt, Stefan Feuerriegel, and Mrinmaya Sachan. 2024b. Automated knowledge concept annotation and question representation learning for knowledge tracing. *arXiv preprint arXiv:2410.01727*.

Zachary A Pardos, Mingyu Feng, Neil T Heffernan, and Cristina Linquist-Heffernan. 2007. Analyzing fine-grained skill models using bayesian and mixed effects methods. *Educational Data Mining*, page 50.

Zachary A Pardos, Neil T Heffernan, Brigham Anderson, Cristina L Heffernan, and Worcester Public Schools. 2006. Using fine-grained skill models to fit student performance with bayesian networks. In *Workshop in Educational Data Mining held at the 8th International Conference on Intelligent Tutoring Systems. Taiwan*.

Rafaella Sampaio de Alencar, Mehmet Arif Demirtas, Adittya Soukarjya Saha, Yang Shi, and Peter Brusilovsky. 2025. Integrating expert knowledge with automated knowledge component extraction for student modeling. In *Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, pages 307–312, New York, NY, USA. ACM.

Alexander Scarlatos, Ryan S Baker, and Andrew Lan. 2025. Exploring knowledge tracing in tutor-student dialogues using llms. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pages 249–259.

Yang Shi, Robin Schmucker, Keith Tran, John Bacher, Kenneth Koedinger, Thomas Price, Min Chi, and Tiffany Barnes. 2024. The knowledge component attribution problem for programming: Methods and tradeoffs with limited labeled data. *Journal of Educational Data Mining*, 16(1):1–33.

Glen Smith, Adit Gupta, and Christopher MacLellan. 2024. Apprentice tutor builder: A platform for users to create and personalize intelligent tutors. *arXiv preprint arXiv:2404.07883*.

Megha Srivastava and Noah Goodman. 2021. Question generation for adaptive education. *arXiv preprint arXiv:2106.04262*.

John Stamper and Zachary A Pardos. 2016. The 2010 kdd cup competition dataset: Engaging the machine learning community in predictive learning analytics. *Journal of Learning Analytics*, 3(2):312–316.

John C Stamper, Kenneth R Koedinger, Ryan SJ d Baker, Alida Skogsholm, Brett Leber, Sandy Demi, Shawnwen Yu, and Duncan Spencer. 2011. Datashop: A data repository and analysis service for the learning science community (interactive event). In *International Conference on Artificial Intelligence in Education*, pages 628–628. Springer.

Jonathan L Templin and Robert A Henson. 2006. Measurement of psychological disorders using cognitive diagnosis models. *Psychological methods*, 11(3):287.

Khushboo Thaker, Peter Brusilovsky, and Daqing He. 2019. Student modeling with automatic knowledge component extraction for adaptive textbooks. In *iTextbooks@ AIED*, pages 95–102.

Yumou Wei, Paulo Carvalho, and John Stamper. 2025. Kcluster: An llm-based clustering approach to knowledge component discovery.

Hanbo Xie, Hua-Dong Xiong, and Robert C Wilson. 2025. Rethinking think-aloud in the age of language models.

Zhang Xiong, Haoxuan Li, Zhuang Liu, Zhuofan Chen, Hao Zhou, Wenge Rong, and Yuanxin Ouyang. 2024. A review of data mining in personalized education: Current trends and future prospects. *Frontiers of Digital Education*, 1(1):26–50.

Jifan Yu, Mengying Lu, Qingyang Zhong, Zijun Yao, Shangqing Tu, Zhengshan Liao, Xiaoya Li, Manli Li, Lei Hou, Hai-Tao Zheng, and 1 others. 2023. Moocradar: A fine-grained and multi-aspect knowledge repository for improving cognitive student modeling in moocs. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2924–2934.

Jifan Yu, Yuquan Wang, Qingyang Zhong, Gan Luo, Yiming Mao, Kai Sun, Wenzheng Feng, Wei Xu, Shulin Cao, Kaisheng Zeng, and 1 others. 2021. Mooccubex: a large knowledge-centered repository for adaptive learning in moocs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4643–4652.

Yongan Yu, Mengqian Wu, Yiran Lin, and Nikki G Lobczowski. 2025. Think: Can large language models think-aloud? *arXiv preprint arXiv:2505.20184*.

A RQ1: KC Extraction

This appendix provides supplementary materials for RQ1 (KC Extraction), including the complete KC example, extraction prompts, matching metrics, and qualitative analysis of granularity differences.

A.1 Complete KC Representation Example

Running Example: Complete Multi-Faceted KC Structure

Problem: Janice walks up 3 flights of stairs 5 times and down 3 flights 3 times in a day. How many flights does she walk total?

Layer 0 KC (High-level concept):

- **Cognitive action**: calculate
- **Content element**: The procedure for calculating a total quantity by multiplying a rate or quantity per unit by the number of units
- **Learning outcome**: Ability to calculate a total quantity by multiplying a rate or quantity per unit by the number of units
- **Relevant think-aloud parts**: "To find the total flights walked up, I need to CALCULATE the product of the number of flights per trip and the number of times she goes up. Number of flights up = (flights per trip) * (number of times going up)"

Layer 1 Sub-KCs (Decomposition):
This Layer 0 KC decomposes into three Layer 1 sub-KCs:

1. **Sub-KC 1**:
   - **Cognitive action**: identify
   - **Content element**: Identifying the rate (flights per trip) and number of units (number of trips)
   - **Learning outcome**: Ability to extract rate and unit count from problem statements
   - **Relevant think-aloud parts**: "I can see that she walks 3 flights per trip"
2. **Sub-KC 2**:
   - **Cognitive action**: model
   - **Content element**: Modeling the multiplication expression using identified quantities
   - **Learning outcome**: Ability to construct mathematical expressions from word problems
   - **Relevant think-aloud parts**: "Number of flights up = (flights per trip) * (number of times going up)"
3. **Sub-KC 3**:
   - **Cognitive action**: calculate
   - **Content element**: Computing the numerical product
   - **Learning outcome**: Ability to execute multiplication operations accurately
   - **Relevant think-aloud parts**: "So that would be 3 flights * 5 times = 15 flights"

This example demonstrates how CogKC's hierarchical structure breaks down a complex procedure (calculating total quantity) into atomic steps with distinct cognitive actions (identify, model, calculate), each grounded in specific think-aloud protocol excerpts.

A.2 TIMSS Cognitive Framework

The TIMSS Mathematics Cognitive Framework (Mullis et al., 2021) is designed specifically for assessing mathematical thinking in international large-scale education studies. It categorizes cognitive processes into three major domains: *knowing*, *applying*, and *reasoning*, with each domain further divided into specific cognitive actions.

In our work, we adopt this classification as the foundation for defining cognitive actions in knowledge components (KCs). To enhance clarity and disambiguate subtle differences between cognitive actions, we slightly modify the original TIMSS descriptions through paraphrasing. This allows the extracted KCs to more precisely reflect the intended cognitive demands of mathematical problem-solving.

We employ this categorization in the prompting process described in the following subsections (Stages 1-4). The cognitive actions associated with each domain are summarized in Table 5, with detailed definitions provided below. In the subsequent stage prompts (Sections A.3-A.6), we use the placeholder "*[TIMSS Cognitive Framework definitions inserted here...]*" to indicate where these definitions are included in the actual prompts provided to the LLM, avoiding repetitive display of the same content across multiple prompt examples.

Domain	Cognitive Actions
Knowing	recall, identify, categorize, calculate, extract, quantify
Applying	select, model, execute
Reasoning	decompose, synthesize, assess, infer, extrapolate, validate

Table 5: TIMSS Cognitive Framework domains and their associated cognitive actions

A.2.1 Detailed Cognitive Action Definitions

The following definitions are used consistently across all stages of the KC extraction pipeline:

Knowing Domain

- **recall**: Retrieve mathematical facts, formulas, or concepts from memory without external aids
- **identify**: Recognize specific mathematical patterns, structures, or properties when presented
- **categorize**: Systematically group or order mathematical objects based on common properties
- **calculate**: Execute well-defined mathematical algorithms and procedures to arrive at results
- **extract**: Select relevant information from graphs, tables, diagrams, or other representations
- **quantify**: Determine the magnitude of physical or abstract attributes using appropriate measurement tools and units

Applying Domain

- **select**: Analyze problem requirements to choose the most effective solution method or strategy
- **model**: Transform real situations or problems into mathematical representations (equations, graphs, diagrams)
- **execute**: Systematically apply selected solution strategies in problem contexts to obtain answers

Reasoning Domain

- **decompose**: Break down complex mathematical situations into smaller, manageable components to understand relationships
- **synthesize**: Combine multiple mathematical ideas, concepts, or approaches to create integrated solutions
- **assess**: Judge the validity or efficiency of solutions, strategies, or arguments based on mathematical criteria
- **infer**: Reach mathematically justified conclusions through logical reasoning from given information
- **extrapolate**: Extend rules or principles discovered in specific cases to more general situations
- **validate**: Logically prove the correctness and validity of conclusions or solutions through mathematical reasoning

A.3 Stage 1: Think-Aloud Generation

In the first stage, we generate a detailed think-aloud solution t from the given problem p. Problem, solution, and figure can each appear as text or image; we label images accordingly.

Think-Aloud Generation Prompt

```
You are an expert mathematician working
through this problem step-by-step. Generate a
genuine think-aloud that shows your internal
problem-solving process as you work toward the
solution.
Problem: {problem_text}
{figure_description}
Question Image: First Figure
Model Solution: {solution_text}
Solution Image: Second Figure (Optional)
Think aloud naturally as you work through
the problem, incorporating these mathematical
thinking aspects and EXPLICITLY mentioning
them in UPPERCASE:
[TIMSS Cognitive Framework definitions
inserted here - see Appendix A.2.1 for
complete list of 15 cognitive actions across
Knowing, Applying, and Reasoning domains]
Example style with explicit mathematical
thinking aspects:
"Okay, let me think through this problem step
by step. I need to IDENTIFY what information
I have and EXTRACT the key details from the
problem statement. So I can see that...
Now I need to SELECT an appropriate solution
strategy. I think the best approach here is
to... Let me DECOMPOSE this into smaller parts:
first I'll... then I'll...
Let me CALCULATE this step: ... = ... Now I
need to SYNTHESIZE these results to...
Finally, let me VALIDATE my answer by checking
if it makes sense with the original problem..."
Important Instructions:

  1. Write ONLY the think-aloud process - no
     introductions or explanations about the
     task

  2. Use first-person language showing your
     thinking ("I need to...", "Let me...")

  3. Show genuine problem-solving steps

  4. Ensure your reasoning leads logically to
     the provided solution

  5. Be natural and authentic in your
     mathematical thinking process
```

6. EXPLICITLY mention the mathematical thinking aspects in UPPERCASE when you perform them

7. Use the exact terms from the definitions above (RECALL, IDENTIFY, CALCULATE, etc.)

8. Integrate these terms naturally into your thinking process, don't force them

A.4 Stage 2: Extract Structured Reasoning Steps

From the think-aloud transcripts extracted in Stage 1, we identify individual problem-solving steps and group them accordingly. In mathematical problem solving, it is common for multiple steps to involve similar or identical knowledge components (KCs). To reduce redundancy and facilitate more efficient KC extraction, we group such steps together.

Solution Steps Extraction Prompt

Input:

- Think-Aloud Process: {think_aloud}
- Problem Context: {problem_context}

Your Task:
Extract the main solution steps from this mathematical think-aloud process, and identify which steps should be grouped together for knowledge extraction.
CRITICAL REQUIREMENT: Complete Think-Aloud Coverage
You MUST ensure that EVERY sentence, phrase, and protocol from the think-aloud process is captured and referenced in the solution steps. No part of the think-aloud should be left unused.
Guidelines:

1. **Comprehensive Coverage**: Read through the entire think-aloud text and ensure every sentence/phrase is assigned to at least one step
2. **Step Identification**: Identify each distinct phase in the problem-solving process
3. For each step:
   - Provide a clear description of what's happening
   - **MANDATORY**: Include ALL relevant parts from the think-aloud text as direct quotes
   - Include even transitional phrases, confirmations, and intermediate thoughts
   - Note which previous steps it depends on (if any)
4. **Granularity Balance**: Focus on meaningful process phases while ensuring complete coverage
5. Keep steps sequential and maintain their logical dependencies
6. **Verification**: Before finalizing, verify that every sentence from the think-aloud appears in at least one step's relevant_think_aloud_parts
7. IMPORTANT: Mark steps that should be grouped together for knowledge extraction
   - Steps that evaluate similar operations or concepts should be given the same group letter
   - Steps that simply continue or confirm a previous step should be grouped with that step
   - For example, steps that evaluate different options but use the same mathematical concepts should be grouped
   - Groups should be identified with letters (A, B, C, etc.)

Expected Output Format:
Return a JSON object with solution steps and their groups. Steps with the same group letter will be analyzed together for knowledge component extraction.
IMPORTANT: The "relevant_think_aloud_parts" arrays must collectively contain EVERY sentence/phrase from the original think-aloud text. Use direct quotes exactly as they appear in the think-aloud.

```
{
  "solution_steps": [
    {
      "step_number": 1,
     "step_description": "Clear description of this step",
      "relevant_think_aloud_parts": [
        "Exact quote from think-aloud sentence 1",
        "Exact quote from think-aloud sentence 2",
        "Any transitional phrases or confirmations"
      ],
      "depends_on_steps": [step numbers this depends on],
      "kc_extraction_group": "A"
    },
    {
      "step_number": 2,
      "step_description": "Description of next step",
      "relevant_think_aloud_parts": [
        "All remaining unassigned quotes",
        "Minor phrases like 'Let me check'"
      ],
      "depends_on_steps": [1],
      "kc_extraction_group": "A"
    }
  ]
}
```

A.5 Stage 3: Extract Preliminary KCs from Reasoning Steps

For each step group extracted in Stage 2, we extract the corresponding knowledge components (KCs). The prompt below includes comprehensive guidelines and illustrative examples for formulating content elements and learning outcomes.

KC Extraction Prompt

CRITICAL: Semantic-First Approach
The Solution Step Description is your PRIMARY guide for KC extraction. It describes the cognitive actions and knowledge types involved in this step. Your task is to:

1. **Identify cognitive actions** mentioned or implied in the step_description (e.g., "calculates" → calculate, "identifies" → identify, "selects strategy" → select)
2. **Extract knowledge components** that support these cognitive actions
3. **Use think-aloud parts** as evidence to validate and refine your KCs
4. **Ensure semantic alignment** between step purpose and extracted KCs

The step_description defines WHAT cognitive work is being done. The think-aloud parts show HOW it's executed.
Input:

- Solution Step {step_number}: {step_description}
- Relevant Think-Aloud Parts: {relevant_think_aloud_parts}
- Problem Context: {problem_context}

What is a Knowledge Component?
A Knowledge Component (KC) represents the intersection of a cognitive action with a specific content element. It describes what students should be able to do (cognitive action) with what they know (content element), forming a discrete unit of competency that can be learned, taught, and assessed. Each KC should be:

- GENERALIZABLE: Applicable across different problems
- ATOMIC: Representing a single, coherent piece of knowledge
- ASSESSABLE: Can be evaluated whether a student has mastered it

Your Task:
Extract Knowledge Components that enable the cognitive actions described in this solution step.
Step-by-Step Process:

1. **Analyze the step_description** to identify:
   - What cognitive actions are being performed (calculate, identify, decompose, etc.)
   - What mathematical knowledge/concepts are being applied
   - What type of reasoning or problem-solving is occurring
2. **Map to KC cognitive actions** using these indicators:
   - "calculates/computes" → calculate
   - "identifies/recognizes" → identify
   - "selects strategy/chooses approach" → select
   - "decomposes/breaks down" → decompose
   - "synthesizes/combines" → synthesize
   - "makes inference/concludes" → infer
   - "validates/checks" → validate
3. **Extract content elements** that support the cognitive actions
4. **Find supporting evidence** in think-aloud parts
5. **Ensure each KC reflects the step's semantic purpose**

Guidelines for Good KCs:
DO:

- Focus on underlying mathematical principles, not problem-specific details
- Use general mathematical terms instead of specific values
- Include both conceptual knowledge and procedural skills
- Ensure each KC has supporting evidence from the think-aloud text
- Keep cognitive action separate from content element

DON'T:

- Include problem-specific numbers or context
- Create overly broad KCs that combine multiple skills
- Create overly narrow KCs that only apply to this specific problem
- Omit supporting evidence from the think-aloud
- Mix cognitive actions into content element descriptions

Content Element Guidelines:
The content element should:

- Represent core mathematical knowledge or concept
- Be independent of any cognitive action
- Focus on WHAT is learned, not HOW it's used
- Be expressible as a noun phrase

Learning Outcome Guidelines:
The learning outcome should:

- Always begin with “Ability to...”
- Describe what the student can demonstrate after mastering the KC
- Incorporate both the cognitive action and content element
- May represent internal processes not directly observable
- Focus on the capability, which could be either a process or an end result

Examples of Well-Formed Content Elements:

- “The Pythagorean theorem relating the sides of a right triangle”
- “The concept of place value in the decimal number system”
- “The relationship between derivatives and slopes of functions”
- “The properties of parallel lines in Euclidean geometry”
- “The definition of fractions as parts of a whole”

Cognitive Action Categories:
Choose ONE cognitive action for each KC from these categories:
[TIMSS Cognitive Framework definitions inserted here - see Appendix A.2.1 for complete list of 15 cognitive actions across Knowing, Applying, and Reasoning domains]
Semantic Validation Check:
Before finalizing your KCs, verify:

- Each KC directly supports a cognitive action mentioned in the step_description
- Think-aloud evidence actually demonstrates the cognitive action + content element
- KCs collectively represent the step’s problem-solving purpose
- No think-aloud fragments are used that contradict the step’s semantic intent

Step-Grounded Examples:
EFFECTIVE (Semantically Aligned):
Step: “The solver calculates the total spoons and makes an inference about sequence”
KCs:

```
{
  "cognitive_action": "calculate",
  "content_element": "Addition of quantities",
 "learning_outcome": "Add quantities accounting for usage",
  "relevant_think_aloud_parts": ["CALCULATE: 12+3=15"]
},
{
  "cognitive_action": "infer",
  "content_element": "Temporal sequence of operations",
  "learning_outcome": "Reason about event order",
  "relevant_think_aloud_parts": ["INFERence based on
                                  sequence of events"]
}
```

INEFFECTIVE (Semantically Misaligned):
Step: “The solver calculates the total spoons”
KC:

```
{
  "cognitive_action": "identify",
  "content_element": "Unknown quantities",
  "relevant_think_aloud_parts": ["I don't know how many"]
}
```

Problem: Step is about calculation, but KC focuses on identification from different part of process.
Prerequisites:
After identifying KCs, determine if they have prerequisite relationships. A prerequisite is knowledge that must be acquired before understanding another concept.
Important: The values for prerequisite_kc and dependent_kc MUST be the 0-based indices of the corresponding knowledge components within the knowledge_components list in THIS JSON response. For example, 0 refers to the first component in the list, 1 refers to the second, and so on.
Output Format:

```
{
  "knowledge_components": [
    {
      "cognitive_action": "action",
      "content_element": "knowledge/skill description",
      "learning_outcome": "outcome of mastering KC",
      "relevant_think_aloud_parts": ["quote from
                                      think-aloud"]
    }
  ],
  "prerequisites": [
    {"prerequisite_kc": 0, "dependent_kc": 1,
     "reason": "prerequisite reason"}
  ]
}
```

A.6 Stage 4: Recursively Decompose KCs

In Stage 4, we evaluate whether each KC extracted in Stage 3 can be further decomposed into more atomic sub-components. The prompt includes guidelines for constructing valid content elements and learning outcomes, along with examples and references to the TIMSS cognitive framework.

Recursive KC Decomposition Prompt

CRITICAL: Maintain Semantic Context
This KC was extracted from a specific solution step with a defined cognitive purpose. When breaking it down, you must:

1. **Preserve the step’s semantic intent** - sub-components should support the original step’s cognitive work
2. **Stay within the step’s scope** - don’t create sub-components that belong to different solution steps
3. **Maintain cognitive coherence** - sub-components should logically decompose the parent KC’s cognitive action

Current KC Details:

- ID: {kc_id}

- Cognitive Action: {cognitive_action}
- Content Element: {content_element}
- Learning Outcome: {learning_outcome}
- Step Context: {step_context}
- Relevant Think-Aloud Parts: {relevant_think_aloud_parts}

Your Task:
Analyze whether this KC can be broken down into more atomic sub-components that **support the same solution step's cognitive work**.
What is a Knowledge Component?
A Knowledge Component (KC) represents the intersection of a cognitive action with a specific content element. It describes what students should be able to do (cognitive action) with what they know (content element), forming a discrete unit of competency that can be learned, taught, and assessed.
Key Questions to Consider:

1. Is this KC truly atomic (a single, indivisible skill/knowledge)?
2. Can it be split into sequential steps or parallel skills?
3. Would breaking it down help in learning/assessment?

About the "is_atomic" Field:
Set "is_atomic" to:

- TRUE only if the KC truly cannot be meaningfully broken down further into separate knowledge/skill components
- FALSE if the KC can be broken down into smaller components (even if you're not providing them)

Make your decision carefully. If a KC could theoretically be broken down but you're not providing sub-components, still mark "is_atomic" as FALSE.
If Breaking Down is Appropriate:
Identify one of these breakdown types:

- Sequential steps: If the KC represents a process with ordered steps
- Parallel skills: If the KC combines multiple independent abilities

Guidelines for Sub-Components:

1. Each sub-component should be GENERALIZABLE across problems
2. Each must have supporting evidence from the parent's think-aloud text
3. Use the same cognitive action categories as below
4. Maintain the same level of generality as the parent KC
5. Keep cognitive action separate from content element

Content Element Guidelines:
The content element should:

- Represent core mathematical knowledge or concept
- Be independent of any cognitive action
- Focus on WHAT is learned, not HOW it's used
- Be expressible as a noun phrase

Learning Outcome Guidelines:
The learning outcome should:

- Always begin with "Ability to..."
- Describe what the student can demonstrate after mastering the KC
- Incorporate both the cognitive action and content element
- May represent internal processes not directly observable
- Focus on the capability, which could be either a process or an end result

Examples of Well-Formed Content Elements:

- "The Pythagorean theorem relating the sides of a right triangle"
- "The concept of place value in the decimal number system"
- "The relationship between derivatives and slopes of functions"
- "The properties of parallel lines in Euclidean geometry"
- "The definition of fractions as parts of a whole"

Cognitive Action Reference:
You MUST choose ONE cognitive action for each KC from these categories:
[TIMSS Cognitive Framework definitions inserted here - see Appendix A.2.1 for complete list of 15 cognitive actions across Knowing, Applying, and Reasoning domains]
Prerequisites:
After identifying sub-components, determine if they have prerequisite relationships. A prerequisite is knowledge that must be acquired before understanding another concept. Important: The values for prerequisite_kc and dependent_kc MUST be the 0-based indices of the corresponding knowledge components within the knowledge_components list in THIS JSON response. For example, 0 refers to the first component in the list, 1 refers to the second, and so on.
Output Format:

```
{
  "is_atomic": true/false,
 "breakdown_type": "sequential_steps"|"parallel_skills"|null,
  "knowledge_components": [
    {
      "cognitive_action": "action",
      "content_element": "knowledge/skill description",
      "learning_outcome": "outcome of mastering KC",
      "relevant_think_aloud_parts": ["quote"]
    }
  ],
  "prerequisites": [
    {"prerequisite_kc": 0, "dependent_kc": 1,
     "reason": "prerequisite reason"}
  ]
}
```

Example of Good Breakdown:

```
{
  "is_atomic": false,
  "breakdown_type": "sequential_steps",
  "knowledge_components": [
    {
      "cognitive_action": "identify",
      "content_element": "Derivatives/slopes relation",
      "learning_outcome": "Identify derivative = slope",
      "relevant_think_aloud_parts": ["To find slope,
        substitute x into derivative"]
    },
    {
      "cognitive_action": "calculate",
      "content_element": "Evaluating derivative",
      "learning_outcome": "Calculate slope at point",
      "relevant_think_aloud_parts": ["Slope = 1-0 = 1"]
    }
  ],
  "prerequisites": [
    {"prerequisite_kc": 0, "dependent_kc": 1,
     "reason": "Understand relation before evaluating"}
  ]
}
```

A.7 KC Merging: Formal Specification

This section provides details and prompts for our two-stage KC merging process described in Section 3.

A.7.1 Notation and Definitions

Let $K = \{k_1, k_2, \ldots, k_n\}$ denote the set of extracted KCs. For each KC k_i, we compute an embedding $\mathbf{e}_i = \text{Encode}(k_i)$ using Sentence-Transformer (`all-MiniLM-L6-v2`), where the input text concatenates the content element, learning outcome, and relevant think-aloud parts. Candidate pairs are identified using cosine similarity $\text{sim}(k_i, k_j) = \cos(\mathbf{e}_i, \mathbf{e}_j)$.

A.7.2 Stage 1: Layer 0 Merging

After Stage 3 (KC extraction), we perform pairwise comparison of all Layer 0 KCs. For each pair (k_i, k_j) where $\text{sim}(k_i, k_j) > \tau_1$ (with $\tau_1 = 0.7$), an LLM evaluates three criteria on a 0–2 scale:

1. **Principle** (s_p): Do both KCs rely on the same fundamental mathematical principle or concept? Do they apply identical reasoning that would yield the same results?
2. **Consistency** (s_c): Do both KCs produce identical results across all equivalent conditions? Do they apply universally to the same range of problems?
3. **Transferability** (s_t): Does mastery of one KC directly lead to mastery of the other without additional instruction? Is the transfer bidirectional?

The total score is $S = s_p + s_c + s_t \in [0, 6]$. KCs merge if $S \geq 5$:

$$\text{merge}(k_i, k_j) = \mathbf{1}[S(k_i, k_j) \geq 5] \qquad (1)$$

Merges are tracked using Union-Find to handle transitive relationships: if k_i merges with k_j and k_j merges with k_l, then all three belong to the same equivalence class.

A.7.3 Layer 0 Merging Prompt

The following prompt is used in Stage 1 to evaluate whether two similar KCs should be merged based on the Principle, Consistency, and Transferability criteria.

Task: These two Knowledge Components were extracted from different problem-solving contexts. Evaluate whether they denote a similar KC based on the criteria below.
Scale: [0] = Does not meet the criterion, [1] = Partially meets, [2] = Fully meets
Criteria:

1. **[Principle]** Do the two KCs rely on the same fundamental principle or concept?
   - Do both KCs apply the exact same mathematical property, rule, or theorem as an essential part of their reasoning?
   - Can the method from one KC be applied directly to solve the same type of problem in the other KC?
2. **[Consistency]** Do the two KCs consistently produce the same results across different contexts?
   - Do both KCs yield identical results under all equivalent conditions?
   - Do both KCs apply universally to the same range of problems?
3. **[Transferability]** If a student masters one skill, will they naturally excel at the other?
   - Does proficiency in one skill directly lead to proficiency in the other without additional instruction?
   - Is the transfer bidirectional?

KC Descriptions:
[Two KC descriptions with cognitive action, content element, learning outcome, and think-aloud parts]
Output Format:

```
{
  "should_merge": true/false,
  "score": {"principle": 0-2, "consistency": 0-2,
            "transferability": 0-2},
  "reason": {"principle": "...", "consistency": "...",
             "transferability": "..."},
  "merged_kc": {...} // if should_merge is true
}
```

If total score $\geq$ 5, merge the KCs into a generalized representation.

A.7.4 Stage 2: Hierarchical Merging

After Stage 4 (decomposition), we apply clustering-based merging for scalability. Using FAISS (Douze et al., 2024) with k-NN search ($k = 10$, threshold $\tau_2 = 0.5$), we form initial clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$.

For each cluster C with $|C| > 1$, an LLM classifies the relationship:

$$f(C) \in \{\text{equivalence}, \text{hierarchy}, \text{split}\} \quad (2)$$

- **Equivalence**: All KCs in C represent the same concept $\rightarrow$ merge into single KC
- **Hierarchy**: KCs are related but distinct $\rightarrow$ establish parent-child structure
- **Split**: KCs are sufficiently different $\rightarrow$ keep separate

A.7.5 Hierarchical Merging Prompt

The following prompt is used to analyze clusters of similar KCs identified by the embedding search. The LLM determines one of three relationships for each sub-cluster.

You are an expert in analyzing Knowledge Components (KCs) derived from educational problem-solving steps.
Given a cluster of potentially related KCs below (identified by temporary IDs like kc_0, kc_1, etc.), identify subgroups within this cluster and determine the relationship between the KCs in each subgroup.
Analyze the provided cluster of Knowledge Components (KCs) and identify potential sub-groups within it.
For each identified sub-group, categorize the relationship between its KCs into ONE of the following three cases:
Case 1 (Equivalence): All KCs within this sub-group represent exactly the same underlying concept and should be merged into a single, representative KC.
Case 2 (Hierarchy - Create Parent): The KCs in this sub-group represent distinct but closely related concepts (e.g., components of a larger concept, specific examples of a general principle) that should be grouped under a NEW, broader parent KC. The original KCs in the sub-group will become children of this new parent.
Case 3 (Split): The KCs within this sub-group represent distinct concepts that are sufficiently different and should remain separate KCs. This case also applies to KCs that do not fit into any identified Case 1 or Case 2 sub-group.
Input KCs:
You will be given a list of KCs, each with:

- ID: A temporary unique identifier (e.g., kc_0, kc_1)
- Cognitive Action: The cognitive process involved (e.g., recall, identify, decompose)
- Content Element: The main content describing the concept
- Learning Outcome: What a student should learn from this KC
- Relevant Think-Aloud: Problem-specific reasoning excerpts

Cognitive Action Categories:
Knowing Domain

- **recall**: Retrieve mathematical facts, formulas, or concepts from memory
- **identify**: Recognize specific mathematical patterns or structures
- **categorize**: Systematically group mathematical objects based on properties
- **calculate**: Execute well-defined mathematical algorithms
- **extract**: Select relevant information from representations
- **quantify**: Determine magnitude using appropriate measurement

Applying Domain

- **select**: Choose the most effective solution method
- **model**: Transform situations into mathematical representations
- **execute**: Apply solution strategies to obtain answers

Reasoning Domain

- **analyze**: Break down problems to understand relationships
- **integrate**: Combine concepts or procedures
- **judge**: Evaluate solutions or choose best approaches

- **generate**: Develop original strategies or extend solutions
- **decompose**: Simplify complex problems into manageable parts

Output Format:
Return a JSON object with the following structure:

```
{
  "results": [
    {
      "kc_ids": ["kc_0", "kc_1", ...],
      "case": 1, 2, or 3,
      "explanation": "Reasoning for this decision",
      "merged_kc": {...} or null,
      "parent_kc": {...} or null
    }
  ]
}
```

For Case 1 (Equivalence), provide merged_kc with cognitive_action, content_element, and learning_outcome fields.
For Case 2 (Hierarchy), provide parent_kc with the same fields for the new parent KC.
For Case 3 (Split), both merged_kc and parent_kc should be null.
Input KCs:
[KC details with temporary IDs are dynamically inserted here]

These prompts enable principled decisions about KC relationships while preserving the multi-faceted KC structure (cognitive action, content element, learning outcome) and hierarchical organization from the extraction pipeline.

A.8 Meta-Evaluation of KC Matching Metric

We conducted an annotation study to obtain ground-truth labels for gold-predicted KC pairs, which were subsequently used to validate our automatic metric. Specifically, we randomly sampled 30 gold KCs and, for each, 3 predicted KCs from our approach that exhibited diverse cosine similarities with the gold KC. Then, we did the same for 10 random gold KCs for each of the two baseline methods. This resulted in a total of 50 unique gold KCs and 150 gold-predicted KC pairs.

Participant Recruitment. We recruited 15 annotators via Prolific, all of whom were verified Domain Experts in Mathematics and who all passed a qualification task (age: M=36.5, SD=12.2; gender: 10 male, 5 female; country of residence: 13 UK, 2 US). Participants were required to: (1) be conversationally fluent in English, (2) have advanced understanding of mathematics education concepts, and (3) be able to understand and evaluate relationships between educational concepts. Each annotator evaluated 10 gold KCs and their relevant predicted KCs (i.e., 30 gold-predicted pairs), spending approximately 30 minutes and receiving compensation of £11. Participants were informed that data would be used for research purposes only and would not be shared with third parties; informed consent was obtained before participation.

Annotation Interface. For each gold-predicted KC pair, annotators selected one of four relationship labels based on semantic meaning:

- **Equivalent**: The Main KC and Similar KC describe the same concept or skill, even if they use different wording or level of detail.
- **Main KC is higher-level**: The Main KC describes a more general or superordinate concept that encompasses the Similar KC.
- **Main KC is lower-level**: The Main KC describes a more specific or subordinate concept that is part of the Similar KC.
- **Not related**: The Main KC and Similar KC describe different concepts or skills that do not share a semantic relationship.

Annotators were informed that submissions would be manually verified and may be rejected if found to be incorrect. Each gold-predicted KC pair was annotated by three annotators and we kept pairs where a majority of annotators agreed on a label (126 pairs, 84% of pairs), forming the final dataset used to validate our metric.

A.9 KC Granularity Analysis

This section provides detailed qualitative examples illustrating the granularity differences between our hierarchical KC extraction framework and baseline methods, as evidenced by the relationship label distributions in Table 2.

A.9.1 Methodological Note on KC Representation

Our framework extracts KCs in a structured hierarchical format with multiple components (content elements, learning outcomes, prerequisite relationships), whereas baseline methods and gold standard KCs use single-string descriptions. For fair comparison with the ATC gold standard, we extracted only the *content element* and *learning outcome* components from our structured KCs. This representation difference affects cosine similarity scores: our longer, detailed descriptions yield lower similarity values even when semantically aligned with

gold KCs, contributing to the precision differences observed in Table 1.

A.9.2 Pattern 1: Atomic vs. Application-Level Descriptions

Our framework systematically extracts atomic, procedural knowledge components, while baselines and gold standards often capture broader, application-level competencies.

Example 1: Unit Rate Calculation

- **Gold KC:** "Solve unit rate problems including those involving unit pricing and constant speed. For example, if it took 7 hours to mow 4 lawns, then at that rate, how many lawns could be mowed in 35 hours? At what rate were lawns being mowed?"
- **CogKC-ML1:** "Steps for calculating unit rate. *Ability to execute the necessary steps to calculate a unit rate from given information.*"
- **KCQRL:** "Calculating Unit Rates: A unit rate simplifies a rate by expressing one of the quantities as a single unit. For instance, if a car travels 100 miles in 2 hours, its unit rate is 50 miles per hour. To find a unit rate, divide the first quantity by the second quantity. This concept is essential for standardizing comparisons and solving problems involving proportional relationships efficiently."

Relationship: *high* (gold KC broader than ours)
Analysis: The gold KC describes a complete problem-solving competency with contextualized examples. Our framework decomposes this into the underlying procedural skill ("steps for calculating"), representing a more atomic diagnostic unit. KCQRL provides a detailed conceptual explanation but still operates at the integrated skill level. For adaptive tutoring systems requiring precise sub-skill diagnosis, our atomic formulation enables finer-grained tracking of student progress through the constituent steps of unit rate problem-solving.

A.9.3 Pattern 2: Concise Lexical Matching vs. Detailed Semantic Descriptions

LLMKT's shorter KC descriptions facilitate cosine similarity matching with similarly concise gold standard nodes, while our detailed descriptions yield lower similarity scores despite semantic alignment.

Example 2: Counting Data Observations

- **Gold KC:** "Reporting the number of observations."
- **LLMKT:** "Total Data Points. *Determine the total number of items or observations in a dataset.*"
- **CogKC-ML1:** "The total count of items satisfying a specific condition. *Ability to quantify the total number of items that satisfy a specific condition from a collection of identified data points.*" (cosine similarity: 0.49)
- **KCQRL:** "This refers to the process of recognizing and counting individual data entries displayed on a line plot. Each 'X' or mark above the number line signifies a single data point. Correctly identifying all data points is crucial for determining the total number of observations in a dataset, which is often needed for calculations." (cosine similarity: 0.49)

Relationship: *high* (gold KC broader than ours)
Analysis: LLMKT's concise phrasing ("Total Data Points") achieves higher cosine similarity with the equally concise gold KC through lexical overlap. Our method and KCQRL extract more detailed descriptions that specify the cognitive process ("process of recognizing and counting," "quantify items satisfying a condition"), resulting in lower cosine similarity (0.49) despite capturing semantically equivalent or more detailed knowledge. This illustrates how representation format—not semantic content—drives precision differences in Table 1.

A.9.4 Pattern 3: Specific Adjacent Concepts vs. Broad Standard Competencies

Our framework sometimes extracts specific but conceptually adjacent sub-skills to the gold KC, reflecting the fine-grained decomposition of expert-defined competencies.

Example 3: Fraction Representation

- **Gold KC:** "Understand a fraction a/b with $a > 1$ as a sum of fractions $1/b$. Grade 4 expectations in this domain are limited to fractions with denominators 2, 3, 4, 5, 6, 8, 10, 12, and 100."
- **CogKC-ML1:** "The conceptual representation of a fraction (a/b) on a number line,

where unit intervals are divided into 'b' equal parts and 'a' parts are counted from zero. *Ability to recall how the definition of a fraction translates to its representation on a number line, specifically how unit intervals are divided and parts are counted.*"

Relationship: *high* (gold KC broader than ours)

Analysis: The gold KC specifies understanding fractions as sums of unit fractions—a conceptual competency. Our method extracts a related but more specific sub-skill: representing fractions on number lines through interval division and counting. This represents a concrete procedural component that supports the broader conceptual understanding. While adjacent rather than identical, this fine-grained extraction aligns with our framework's goal of identifying atomic diagnostic units that can be independently assessed and targeted in personalized instruction.

B KC Information Density Analysis

This appendix provides detailed validation that our baseline comprehensive variants (LLMKT-FULL, KCQRL-FULL) match or exceed CogKC's information density, ensuring fair comparison and isolating the effect of hierarchical structure.

B.1 Methodology

We measure information content using character-based metrics (chars/KC), which provide more precise quantification than word counts. For each KC representation, we calculate:

- **Chars/KC**: Total characters divided by KC count, measuring information density per knowledge component
- **KCs/Problem**: Mean number of knowledge components extracted per problem
- **Total Chars/Problem**: Total information content per problem

For CogKC, we consistently measure across all four KC facets: cognitive action, content element, learning outcome, and relevant think-aloud excerpt (first instance). This ensures uniform representation across all comparisons.

Note: Due to inherent LLM generation variability, achieving exact information density matching is challenging. We designed comprehensive variants to have similar or more information than CogKC, ensuring that any performance differences cannot be attributed to baselines having insufficient information content.

B.2 ATC Dataset Results (RQ1)

Table 6 shows information content analysis for the ATC dataset (207 problems).

Table 6: Information content analysis on ATC dataset (207 problems). Full variants (LLMKT-FULL, KCQRL-FULL) match or exceed CogKC's information density, while Basic/Orig variants have substantially less information.

Method	Ch/KC	Ratio	KC/P	Ratio
CogKC-Layer1	**286.0**	**1.00×**	**26.5**	**1.00×**
CogKC-Layer0	285.5	1.00×	10.9	2.43×
LLMKT-ORIG	111.4	2.57×	3.6	7.36×
LLMKT-BASIC	101.5	2.82×	26.5	1.00×
LLMKT-FULL	273.7	**1.04×**	26.5	1.00×
KCQRL-ORIG	21.5	13.30×	3.3	8.03×
KCQRL-BASIC	33.8	8.46×	26.5	1.00×
KCQRL-FULL	334.1	**0.86×**	26.5	1.00×

Ch/KC: Chars per KC. KC/P: KCs per Problem.
Ratio>1.0: CogKC more; <1.0: baseline more

B.3 MathDial Dataset Results (RQ2, RQ3)

Table 7 shows information content analysis for the MathDial dataset (2,861 problems).

Table 7: Information content analysis on MathDial dataset (2,861 problems). Full variants match or exceed CogKC's information density (LLMKT-FULL: 16% more, KCQRL-FULL: 5% more).

Method	Ch/KC	Ratio	KC/P	Ratio
CogKC-ML1	**266.0**	**1.00×**	**27.9**	**1.00×**
CogKC-ML0	271.1	0.98×	12.6	2.22×
LLMKT-BASIC	96.9	2.74×	28.3	0.99×
LLMKT-FULL	308.5	**0.86×**	28.3	0.99×
KCQRL-BASIC	30.7	8.67×	27.5	1.02×
KCQRL-FULL	280.6	**0.95×**	27.5	1.02×

Ch/KC: Chars per KC. KC/P: KCs per Problem.
Ratio>1.0: CogKC more; <1.0: baseline more

B.4 Cross-Dataset Summary

Full variants consistently match or exceed CogKC's information density across both datasets: LLMKT-FULL (0.86–1.04× ratio), KCQRL-FULL (0.86–0.95× ratio), with KC counts controlled within 5%. This ensures fair comparison by eliminating information quantity as a confounding factor.

B.5 CogKC Multi-Layer Processing

The ML0 → ML1 transformation demonstrates CogKC's hierarchical refinement: 2.22–2.43× KC expansion while maintaining constant per-KC density (285.5 → 286.0 chars/KC for ATC, 271.1 → 266.0 for MathDial). This represents progressive decomposition of coarse-grained concepts into fine-grained atomic skills grounded in think-aloud protocols, not information compression.

C RQ2: Question Generation

This appendix provides supplementary materials for RQ2 (Personalized Question Generation), including detailed evaluation methodology, example analyses, implementation specifications, and complete prompts used in the pipeline.

C.1 KC Mastery Assessment Prompt

In the second stage, we assess student mastery of each extracted knowledge component (KC) based on the original tutoring conversation. The system analyzes conversation evidence to determine whether the student demonstrates mastery (1), non-mastery (0), or insufficient information (-1) for each KC. This assessment uses the full hierarchical KC structure with cognitive action, content element, and learning outcome specifications.

KC Mastery Assessment Prompt (Full Mode)

You are an expert educational assessor analyzing student understanding in math tutoring conversations.
PROBLEM CONTEXT: {question_context}
CORRECT SOLUTION: {correct_solution}
CONVERSATION SO FAR: {conversation_text}
KNOWLEDGE COMPONENTS (KCs) TO ASSESS: {kcs_text}
COGNITIVE ACTION DEFINITIONS: When assessing mastery, use these precise definitions for each cognitive action. **IMPORTANT: Always write cognitive actions in UPPERCASE when referencing them in your reasoning.**
[TIMSS Cognitive Framework definitions inserted here - see Appendix A.2.1 for complete list of 15 cognitive actions across Knowing, Applying, and Reasoning domains]
IMPORTANT: Focus your assessment on the SPECIFIC cognitive action listed for each KC. For example:

- For "EXTRACT" KCs: Did the student successfully select/pull out the relevant information?
- For "IDENTIFY" KCs: Did the student recognize the correct patterns/structures?
- For "INFER" KCs: Did the student draw correct logical conclusions from the information?

AVOID OVER-INFERENCE PRINCIPLE:
DO NOT make deep inferences about what a student doesn't know based on a single error or limited conversation.
Guidelines:

- Base assessments ONLY on directly observable evidence in the conversation
- Do NOT infer broader conceptual failures from specific contextual errors
- When in doubt, use INSUFFICIENT_INFO rather than NON_MASTERY
- A single mistake does not prove lack of understanding of an entire concept
- Students may know concepts but make contextual or careless errors

Example of over-inference (WRONG):

- Student: "12 - 3 = 9"
- WRONG inference: "Student doesn't understand the concept of quantity"
- CORRECT assessment: "Cannot determine mastery of quantity concept from this contextual error"

COGNITIVE ACTION MISMATCH HANDLING:
CRITICAL: Assess ONLY the specific cognitive action listed in the KC. If the student's error involves a DIFFERENT cognitive action, use INSUFFICIENT_INFO.
Examples of mismatches:

- KC requires "RECALL" but student's error is about "INFER" → INSUFFICIENT_INFO
- KC requires "IDENTIFY" but student's error is about "CALCULATE" → INSUFFICIENT_INFO
- KC requires "EXTRACT" but student's error is about "SYNTHESIZE" → INSUFFICIENT_INFO

IMPORTANT: Do NOT stretch interpretations to force-fit failures into the given KC's cognitive action.
TASK: Analyze the student's mastery of each KC based on the conversation evidence.
MASTERY STATE ASSESSMENT GUIDELINES:

- Mastery State 1 (MASTERY): Student demonstrates clear understanding and correct application of the SPECIFIC KC skill being assessed.
- Mastery State 0 (NON-MASTERY): Student shows confusion, errors, or misconceptions SPECIFICALLY related to the KC being assessed.

- Mastery State -1 (INSUFFICIENT_INFO): Not enough evidence to determine mastery level for the specific KC skill.

USE THINK-ALOUD EXAMPLES FOR GUIDANCE:

- Each KC includes "think_aloud_examples" showing what the KC looks like in practice.
- Use these examples to better understand what student behaviors indicate mastery of each KC.
- Look for similar patterns, language, or reasoning in the student's responses.
- Students demonstrating behaviors similar to the think-aloud examples likely show mastery.

CONFIDENCE SCORE ASSESSMENT GUIDELINES:
Confidence: 0-1 scale indicating certainty of mastery state assessment.
FOR MASTERY (State 1):

- High confidence (0.8-1.0): Clear demonstrations of correct understanding and application
- Medium confidence (0.5-0.7): Generally correct but with minor hesitations or need for prompting
- Low confidence (0.0-0.4): Minimal or unclear evidence of mastery

FOR NON_MASTERY (State 0):

- High confidence (0.8-1.0): Clear evidence of misconceptions, systematic errors, or confusion
- Medium confidence (0.5-0.7): Some errors related to KC, but might be careless mistakes
- Low confidence (0.0-0.4): Weak evidence of KC-specific problems

FOR INSUFFICIENT_INFO (State -1):

- High confidence (0.8-1.0): Definitively no evidence available for this specific KC
- Medium confidence (0.5-0.7): Very limited evidence, unclear what it indicates
- Low confidence (0.0-0.4): Some evidence exists but too ambiguous to interpret

EVIDENCE AND REASONING:

- Evidence: Specific quotes or behaviors that support the assessment.
- Reasoning: Clear explanation of why you assigned this mastery state. **MUST explicitly state:**
  1. Whether there's DIRECT evidence of the specific KC skill in the conversation
  2. Whether the observed behavior matches the KC's cognitive action definition
  3. Whether you're making inferences beyond what's directly observable
- **MANDATORY: Write all cognitive actions in UPPERCASE in your reasoning (e.g., RECALL, EXTRACT, INFER, not recall, extract, infer)**
- **MUST explicitly reference the specific cognitive action definition and whether the student demonstrated that precise skill. Always use UPPERCASE when writing cognitive actions.** For example: "Based on the 'EXTRACT' definition (select relevant information from representations), the student successfully EXTRACTED the numbers 12, 3, and 5 from the word problem" or "Based on the 'INFER' definition (reach justified conclusions through logical reasoning), the student failed to INFER the correct relationship, concluding 12-3=9 instead of 12+3=15"
- If limited evidence: "Insufficient direct evidence to assess [KC]. Student's error could be contextual rather than conceptual."
- If mismatch detected: "Student's error involves [ACTUAL_ACTION] rather than [KC's ACTION], marking as INSUFFICIENT_INFO"

IMPORTANT:

- KC_ID is the unique identifier for the KC.
- Focus on demonstrated understanding of the SPECIFIC KC skill, not overall problem-solving success.
- **DISTINGUISH BETWEEN KC-SPECIFIC ERRORS AND UNRELATED ERRORS:**
  - Mark NON-MASTERY only when errors are directly related to the specific KC being assessed
  - If a student correctly demonstrates the KC skill but makes unrelated errors (e.g., arithmetic mistakes when assessing identification skills), this may still indicate MASTERY of that specific KC
  - Example: For "identify quantities" KC, focus on whether quantities were correctly identified, not whether subsequent calculations were correct
- Use INSUFFICIENT_INFO when:
  - No DIRECT evidence of the specific KC skill in the conversation

- Student uses alternative methods that bypass the KC being assessed
- Student's error involves a different cognitive action than the KC specifies
- The observed failure cannot be DIRECTLY attributed to the KC's specific cognitive action
- You would need to make significant inferences beyond observable evidence
- A single error could have multiple explanations (contextual, careless, or conceptual)
- Mastery of the Knowledge Component (KC) cannot be directly inferred from the given context

CONCRETE ASSESSMENT EXAMPLES:
Example 1: OVER-INFERENCE (WRONG)

- KC: "RECALL" + "The concept of quantity"
- Student behavior: Says "12 - 3 = 9" when asked for total
- WRONG assessment: "Student doesn't RECALL that quantities combine through addition"
- CORRECT assessment: "INSUFFICIENT_INFO - Cannot determine if student can RECALL quantity concept from this single contextual error. Student may understand quantities but misinterpreted the problem context."

Example 2: COGNITIVE ACTION MISMATCH

- KC: "RECALL" + "The concept of quantity"
- Student behavior: Correctly identifies all quantities (12, 3, 5) but misinterprets context (12-3 instead of 12+3)
- Analysis: Student successfully RECALLS quantity concepts. Error is about contextual INFERENCE, not RECALL.
- Assessment: INSUFFICIENT_INFO (different cognitive action than assessed)

Example 3: DIRECT EVIDENCE (CORRECT)

- KC: "EXTRACT" + "Relevant information from word problems"
- Student behavior: "Julia bought x spoons, husband gave 5, she used 3, ended with 12"
- Analysis: Student successfully EXTRACTED all numerical values and relationships from the problem.
- Assessment: MASTERY (direct evidence of EXTRACTION skill)

EVIDENCE STANDARD FOR NON_MASTERY:
To mark NON_MASTERY, you need:

- DIRECT evidence of confusion about the specific KC
- MULTIPLE instances or CLEAR pattern of the same error
- Error that SPECIFICALLY matches the cognitive action being assessed
- Cannot be explained by simpler causes (misreading, careless error, context confusion)

If these criteria aren't met → use INSUFFICIENT_INFO
CRITICAL PRINCIPLES:

- **ALTERNATIVE SOLUTION METHODS AND INSUFFICIENT_INFO:**
  - If a student uses a mathematically correct alternative method that doesn't demonstrate the specific KC being assessed, use INSUFFICIENT_INFO, NOT NON-MASTERY
  - Example: For "multiplication as repeated addition" KC, if student uses 2+2+2+2 instead of 2*4, this is correct math but doesn't show the target KC skill - assess as INSUFFICIENT_INFO
  - Only mark NON-MASTERY when there is clear evidence of confusion, error, or misconception related to the specific KC
  - Correct alternative approaches that bypass the KC should be treated as insufficient evidence, not failure
- **AVOID CASCADING ERROR INFERENCES:**
  - DO NOT infer failure in one KC based on errors in a different KC or wrong final answers
  - Each KC must be assessed independently based on direct evidence of that specific skill
  - Example: If student gets wrong final answer in "2x + 5 = 25", do NOT assume they failed at "subtraction" just because they failed at "division". Despite subtraction being a prerequisite for division conceptually, the student might have correctly done subtraction (25 - 5 = 20) but failed at division (20 ÷ 2), or vice versa. Assess only what you directly observe, not what you assume based on skill hierarchy.
  - Wrong final answers can result from single errors - assess only the KCs you can directly observe being demonstrated
- Analyze the student's mastery by considering the ENTIRE conversation history provided in "CONVERSATION SO FAR". Look for evidence of the specific KC skill throughout the conversation.

OUTPUT FORMAT (valid JSON only):

```
{
  "kc_assessments": [
    {
      "kc_id": "KC_L0_S3_004",
      "mastery_state": 1,
      "confidence": 0.8,
      "reasoning": "Student correctly identified
                    the unknown variable",
      "evidence": "Student said 'let x be the
                   number of spoons Julia bought'"
    }
  ],
  "confidence_score": 0.75
}
```

Content-Element-Only Mode. For baseline comparison, we also implemented a simplified assessment mode that evaluates only content understanding without considering cognitive actions. This mode uses a shortened prompt that focuses purely on whether students understand the mathematical content/concept described, without analyzing how they use or apply that content.

C.2 Question Generation Prompts

In the third stage, we generate personalized follow-up questions targeting the student's identified knowledge gaps. The system creates one comprehensive problem with multiple sub-questions, each designed to address specific KC mastery gaps revealed in the assessment stage. This prompt operates in two modes: CogKC (hierarchical KC-based) and LLMKT (flat KC-based) for comparative evaluation.

Question Generation Prompt (CogKC Mode)

You are an expert tutor creating personalized follow-up questions for a student based on their previous work.
<PreviousMathProblem> {problem_context} **</PreviousMathProblem>**
<StudentKnowledgeComponentMasteryStatus>
The following knowledge component mastery states were inferred from the student's performance on the above problem: {kc_prompt}
<StudentKnowledgeComponentMasteryStatus/>
<Instruction>
TASK: Generate ONE comprehensive problem with targeted sub-questions for identified knowledge gaps.
REQUIREMENTS:

- Identify "target KCs" for assessment:
  - Target ALL KCs with NON_MASTERY status (both parent and sub-component KCs)
  - If a PARENT KC and its SUB-COMPONENT both have NON_MASTERY, target BOTH
  - Prioritize SUB-COMPONENTS for more specific skill gaps, but include relevant PARENT KCs for broader understanding
  - Only ignore KCs with INSUFFICIENT_INFO or MASTERY status
- Generate exactly ONE main problem that provides rich context
- Create one sub-question for each target KC
- Maximum 7 sub-questions (if more than 7 target KCs, prioritize by confidence score)

GENERATION STRATEGY:

1. Review all NON_MASTERY KCs and identify which are "targets" based on the rules above
2. Create a main problem context that naturally allows questions about each target KC
3. Design sub-questions where each precisely targets one specific KC
4. Ensure sub-questions are independent (can be attempted in any order)

CONFIDENCE SCORE INTERPRETATION GUIDELINES:
Confidence: 0-1 scale indicating certainty of mastery state assessment.
FOR MASTERY (State 1):

- High confidence (0.8-1.0): Clear demonstrations of correct understanding and application
- Medium confidence (0.5-0.7): Generally correct but with minor hesitations or need for prompting
- Low confidence (0.0-0.4): Minimal or unclear evidence of mastery

FOR NON_MASTERY (State 0):

- High confidence (0.8-1.0): Clear evidence of misconceptions, systematic errors, or confusion
- Medium confidence (0.5-0.7): Some errors related to KC, but might be careless mistakes
- Low confidence (0.0-0.4): Weak evidence of KC-specific problems

FOR INSUFFICIENT_INFO (State -1):

- High confidence (0.8-1.0): Definitively no evidence available for this specific KC
- Medium confidence (0.5-0.7): Very limited evidence, unclear what it indicates
- Low confidence (0.0-0.4): Some evidence exists but too ambiguous to interpret

MAIN PROBLEM GUIDELINES:

- Create a problem that is structurally similar to the previous problem
- Use the same type of scenario (e.g., if previous was about shopping, keep shopping context)
- Maintain similar complexity level and mathematical operations
- Change only surface details (names, numbers, specific items)
- The main problem serves as familiar context for targeted sub-questions
- Must feel cohesive and natural, not forced

SUB-QUESTION GUIDELINES:

- Each sub-question targets EXACTLY ONE KC
- Label as (a), (b), (c), etc.
- Analyze each KC's description to understand what specific skill or knowledge to assess
- Design questions that isolate and test the specific gap described in each KC
- Question types should match the nature of the skill:
  - Computational skills: Ask for calculations
  - Procedural skills: Ask for step-by-step work
  - Conceptual understanding: Ask for explanations
  - Verification skills: Ask for checking or validation
  - Application skills: Ask to use in context
- Make questions independent - struggling with (a) shouldn't prevent attempting (b)
- If needed for independence, provide necessary values in the sub-question itself

COGNITIVE ACTION DEFINITIONS FOR KC UNDERSTANDING:
[TIMSS Cognitive Framework definitions inserted here - see Appendix A.2.1 for complete list of 15 cognitive actions across Knowing, Applying, and Reasoning domains]
IMPORTANT: When designing sub-questions, consider the SPECIFIC cognitive action each KC requires. For example:

- For "extract" KCs: Create questions that require students to pull out relevant information
- For "identify" KCs: Create questions that require students to recognize patterns or structures
- For "infer" KCs: Create questions that require students to draw logical conclusions

HANDLING SIMILAR KCs:

- Some target KCs may have very similar descriptions (e.g., "compute addition" vs "calculate sum")
- When you encounter similar KCs:
  - Option 1: If they're nearly identical, create ONE sub-question that addresses both and list both KC IDs
  - Option 2: If they have subtle differences, create distinct sub-questions that emphasize those differences
- Example handling:
  - KC1: "Add two-digit numbers"
  - KC2: "Compute sums of two-digit values"
  - Solution: (a) Calculate 45 + 67 [targets KC1]
  - (b) Show your step-by-step process for finding 38 + 54 [targets KC2]

FALLBACK FOR NO NON_MASTERY KCs:
If there are NO KCs with "NON_MASTERY" status, generate a practice problem similar to the previous problem. Use different numbers and slight contextual variations while maintaining the same mathematical structure. Include 2-3 sub-questions that reinforce the demonstrated mathematical skills.
<Instruction/>
OUTPUT FORMAT (valid JSON only):
IMPORTANT: For the 'targeted_kc' field, use ONLY the bracketed KC ID without any descriptive text.

```
{
  "target_kcs": ["List of KC IDs to be targeted"],
  "main_problem": "The comprehensive problem context",
  "sub_questions": [
    {
      "label": "a",
      "text": "Sub-question text",
      "targeted_kc": "KC_ID",
      "rationale": "How this addresses this KC's gap"
    }
  ]
}
```

LLMKT Baseline Mode. For baseline comparison with LLMKT, the prompt is modified to use flat KC representations without hierarchical structure or cognitive action specifications. The LLMKT mode shows only KC name and description fields, while maintaining the same question generation strategy and targeting rules.

Vanilla Mode. We also implemented a vanilla baseline that generates questions through direct

conversation analysis without KC decomposition. This mode analyzes the conversation directly to identify learning gaps and skill areas where the student showed uncertainty, made errors, or needed guidance, then generates targeted questions in a single LLM call.

Question Generation Prompt (Vanilla Mode)

You are an expert tutor creating personalized follow-up questions for a student based on their previous work.
<PreviousMathProblem> {problem_context} **</PreviousMathProblem>**
Based on conversation analysis, identify learning gaps and skill areas where the student showed uncertainty, made errors, or needed guidance. Priority should be given to the most critical gaps that need immediate attention.
<Instruction>
TASK: Generate ONE comprehensive problem with targeted sub-questions for identified knowledge gaps.
REQUIREMENTS:

- Identify "target gaps" for assessment:
  - Target all areas where the student showed uncertainty, made errors, or needed guidance
  - Prioritize the most critical gaps that need immediate attention
  - If no clear gaps exist, focus on reinforcement of demonstrated mathematical skills
- Generate exactly ONE main problem that provides rich context
- Create one sub-question for each target gap
- Maximum 7 sub-questions (if more than 7 target gaps, prioritize by criticality)

GENERATION STRATEGY:

1. Review the conversation and identify which are "targets" based on the rules above
2. Create a main problem context that naturally allows questions about each target gap
3. Design sub-questions where each precisely targets one specific gap or skill
4. Ensure sub-questions are independent (can be attempted in any order)

CONFIDENCE INTERPRETATION GUIDELINES:
Assess certainty of gap identification based on conversation evidence.
FOR CLEAR GAPS (High Evidence):

- Strong evidence: Clear demonstrations of misconceptions, systematic errors, or confusion
- Medium evidence: Some errors or hesitations, but might be careless mistakes
- Weak evidence: Minimal or unclear evidence of specific problems

FOR UNCLEAR AREAS (Low Evidence):

- Weak evidence: Some indication of difficulty but too ambiguous to interpret definitively
- No evidence: Definitively no evidence available for this specific skill area

MAIN PROBLEM GUIDELINES:

- Create a problem that is structurally similar to the previous problem
- Use the same type of scenario (e.g., if previous was about shopping, keep shopping context)
- Maintain similar complexity level and mathematical operations
- Change only surface details (names, numbers, specific items)
- The main problem serves as familiar context for targeted sub-questions
- Must feel cohesive and natural, not forced

SUB-QUESTION GUIDELINES:

- Each sub-question targets EXACTLY ONE gap or skill area
- Label as (a), (b), (c), etc.
- Analyze each gap to understand what specific skill or knowledge to assess
- Design questions that isolate and test the specific gap described in each area
- Question types should match the nature of the skill:
  - Computational skills: Ask for calculations
  - Procedural skills: Ask for step-by-step work
  - Conceptual understanding: Ask for explanations
  - Verification skills: Ask for checking or validation
  - Application skills: Ask to use in context
- Make questions independent - struggling with (a) shouldn't prevent attempting (b)
- If needed for independence, provide necessary values in the sub-question itself

HANDLING SIMILAR GAPS:

- Some target gaps may have very similar descriptions (e.g., "addition errors" vs "calculation mistakes")
- When you encounter similar gaps:
  - Option 1: If they're nearly identical, create ONE sub-question that addresses both
  - Option 2: If they have subtle differences, create distinct sub-questions that emphasize those differences
- Example handling:
  - Gap1: "Addition computation errors"
  - Gap2: "Multi-digit addition process confusion"
  - Solution: (a) Calculate 45 + 67 [targets Gap1]
  - (b) Show your step-by-step process for finding 38 + 54 [targets Gap2]

FALLBACK FOR NO CLEAR GAPS:
If there are NO clear gaps identified from the conversation, generate a practice problem similar to the previous problem. Use different numbers and slight contextual variations while maintaining the same mathematical structure. Include 2-3 sub-questions that reinforce the demonstrated mathematical skills.
<Instruction/>
OUTPUT FORMAT (valid JSON only):
IMPORTANT: For the 'targeted_kc' field, use "CONVERSATION_BASED" as the identifier.

```
{
  "target_kcs": ["List of gap areas to be targeted"],
  "main_problem": "The comprehensive problem context",
  "sub_questions": [
    {
      "label": "a",
      "text": "Sub-question text",
      "targeted_kc": "CONVERSATION_BASED",
      "rationale": "How this addresses this gap"
    }
  ]
}
```

C.3 LLM-as-Judge Evaluation Prompts

For RQ2 evaluation, we employ an LLM-as-judge approach to compare question quality between CogKC and LLMKT methods. The evaluation uses pairwise comparison with position swapping (two evaluation runs with reversed order) to prevent positional bias. The LLM evaluates three key aspects: (1) individual sub-question precision in targeting knowledge gaps, (2) gap identification accuracy, and (3) overall better-set judgment.

Position Randomization. To prevent positional bias, we run each evaluation twice with question sets in opposite positions (LLMKT first vs CogKC first), using neutral labels (Set A, Set B) that hide method identity from the evaluator. Results from both runs are averaged to obtain unbiased assessments.

Pairwise Question Quality Evaluation Prompt

You are an expert math education researcher evaluating AI-generated follow-up questions designed to target specific mathematical competencies revealed through student-tutor conversations. Your task is to assess how well each individual sub-question targets the student's demonstrated competency gaps.
IMPORTANT CONTEXT: The two question sets you will evaluate (QuestionSetA and QuestionSetB below) are ANALOGOUS PROBLEMS designed to test the same mathematical concepts and competencies that were identified as gaps from the original conversation. These question sets use different surface details (different characters, objects, numbers) but target the same underlying mathematical skills and misconceptions revealed in the student-tutor interaction.

<OriginalMathProblem>
{problem_context}
</OriginalMathProblem>

<CorrectSolutionToOriginalProblem>
{ground_solution}
</CorrectSolutionToOriginalProblem>

<StudentTutorConversationAboutOriginalProblem>
{conversation}
</StudentTutorConversationAboutOriginalProblem>

<Instruction>
EVALUATION PHILOSOPHY: "Surgical precision over broad coverage" - Reward questions that efficiently target the student's actual demonstrated gaps rather than generic practice.
IDENTIFIED CRITICAL COMPETENCY GAPS
The following specific mathematical competency gaps have been identified from the conversation:
{gaps_text}
EVALUATION TASK: Use these identified gaps to evaluate how well each question set targets the student's demonstrated competencies.
EVALUATE EACH SUB-QUESTION INDIVIDUALLY
For each sub-question (a, b, c, etc.), evaluate:
Gap Targeting Precision (1-4 scale):

- 4 (Surgical): Directly addresses a specific critical gap with laser focus
- 3 (Targeted): Addresses a relevant gap with good specificity
- 2 (Broad): Addresses related competencies but lacks specificity to demonstrated gaps
- 1 (Unfocused): Generic question that doesn't specifically target demonstrated gaps

Gap Coverage Assessment:

- Which specific competency gap(s) does this question target?
- How critical is this gap for this specific student?
- Explain your reasoning for the precision score

STEP 3: DETERMINE OVERALL WINNER
Based on your evaluation, determine which question set provides superior pedagogical intervention and explain why.
MATHEMATICAL COMPETENCIES TO CONSIDER:
Computational & Procedural:

- Computational accuracy and fluency
- Number sense and quantitative reasoning
- Algorithmic thinking and execution
- Mathematical fact recall and application

Mathematical Reasoning:

- Analyzing mathematical relationships
- Making logical mathematical inferences
- Evaluating mathematical statements and claims
- Recognizing mathematical patterns and structures

Problem-Solving Strategies:

- Decomposing complex mathematical problems
- Selecting appropriate mathematical approaches
- Monitoring mathematical reasoning processes
- Verifying mathematical solutions and their reasonableness

Mathematical Communication:

- Interpreting mathematical language and notation
- Extracting mathematical information from complex descriptions
- Understanding mathematical relationships in context
- Communicating mathematical ideas clearly

EVALUATION PRINCIPLES:

- Reward surgical precision over comprehensive scaffolding when both achieve the learning objectives
- Recognize that fewer, well-targeted questions may be more pedagogically effective than extensive scaffolding
- Value approaches that identify and address root causes rather than symptoms
- Consider cognitive load - excessive scaffolding can hinder rather than help learning
- Value mathematical competencies that transfer across problem types
- Recognize that complex mathematical problems require multiple competencies working together
- Consider both immediate computational needs and long-term mathematical reasoning development
- Assess whether questions develop mathematical understanding vs. only testing procedures
- Evaluate the mathematical sophistication and transferability of competencies targeted
- Consider how competencies support future mathematical learning and problem-solving

Your task is to determine which approach provides superior pedagogical intervention by:

- Identifying the CORE mathematical competencies that, if addressed, would most effectively remediate the student's errors
- Evaluating which approach most efficiently targets these core competencies
- Recognizing that effective intervention may require either comprehensive scaffolding OR surgical precision, depending on the student's needs
- Considering whether additional scaffolding questions add pedagogical value or create unnecessary cognitive load
- Determining which approach is most appropriate for this specific student's demonstrated learning patterns and needs

IMPORTANT: Value question sets that precisely target the student's actual demonstrated needs. If a student needs detailed scaffolding, comprehensive approaches should score highly. If a student needs focused intervention on specific gaps, surgical approaches should score highly. The goal is matching pedagogical approach to identified needs, not preferring one intervention style over another.

</Instruction>

EVALUATION TASK: Below are the two analogous question sets mentioned above. Evaluate how well each set targets the mathematical

```
competency gaps identified from the original
Julia/spoons conversation.

<QuestionSetA>
{set_a_question}
</QuestionSetA>

<QuestionSetB>
{set_b_question}
</QuestionSetB>

<OutputFormat>
Provide your evaluation in the following JSON
format:

{
  "set_a_individual_questions": [
   {"question_label": "a", "gap_targeting_precision": 3.5,
     "targeted_gaps": ["gap_1", "gap_2"],
     "reasoning": "Explanation of precision score"}
  ],
  "set_b_individual_questions": [
   {"question_label": "a", "gap_targeting_precision": 4.0,
     "targeted_gaps": ["gap_1"],
     "reasoning": "Explanation of precision score"}
  ],
  "overall_evaluation": {
    "winner": "set_a" or "set_b" or "tie",
    "winner_reasoning": "Explanation of which set
                         better targets critical gaps"
  }
}

IMPORTANT:

  • Provide individual scores and detailed
    reasoning for each question

  • Focus on educational assessment of how
    well each question targets specific gaps

  • For targeted_gaps, reference specific
    gaps by their gap_id (e.g., "gap_1",
    "gap_2")

  • Do NOT use generic category references
    like "primary_gap" or "secondary_gap"

  • Be specific about which exact gaps each
    question targets

</OutputFormat>
```

C.4 LLM-as-Judge Validation via Human Evaluation

C.4.1 Validation Study Overview

To validate the reliability of our LLM-as-judge evaluation approach used in RQ2, we conducted a comprehensive human evaluation study. Expert educators independently evaluated a sample of generated question sets using the same evaluation framework as the LLM judge. We compared LLM judgments against human expert judgments across three key dimensions: **Gap Targeting Precision** (1-4 scale scoring), **Gap Identification** (which specific knowledge gaps are targeted), and **Better-Set Judgment** (overall quality preference between CogKC and LLMKT).

Participant Recruitment We recruited 6 expert mathematics educators (M=10.5 years teaching experience, SD=2.1; 4 male, 2 female; all residing in Republic of Korea) from an online community for public math school teachers through Google Form. Participants were compensated at 50,000 KRW per hour for their evaluation work. Participants were organized into 3 evaluation sets with 2 raters each. All participants had extensive K-12 teaching experience and familiarity with diagnostic assessment and personalized learning approaches. Participants were informed that data would be used for research purposes only and would not be shared with third parties; informed consent was obtained before participation.

Evaluation Design Each evaluation set contained 7 conversations shared across both raters for inter-rater reliability calculation, yielding 21 conversations across 3 evaluation sets. For each conversation, evaluators assessed generated questions on three dimensions aligned with the LLM judge evaluation framework:

1. **Gap Targeting Precision** (1-4 scale): Evaluators rated each sub-question on how precisely it targets the identified knowledge gap. This enables direct comparison with LLM precision scores via correlation analysis.

2. **Gap Identification**: Evaluators identified which specific knowledge gaps (e.g., gap_1, gap_2) each sub-question targets. This enables comparison with LLM gap identification via Jaccard similarity.

3. **Better-Set Judgment**: Evaluators indicated overall quality preference between CogKC and LLMKT question sets for each conversation. This enables comparison with LLM Better-Set judgments via agreement rate.

Annotation Interface. Evaluators completed assessments via Google Forms (originally in Korean). Each evaluation page presented:

1. **Context**: Original math problem, correct solution, and student-tutor conversation transcript.

2. **Knowledge Gap Quality** (5-point scales with endpoint anchors):

   - Comprehensiveness: 1 = "Important gaps missing" to 5 = "All gaps covered"

- Precision: 1 = "Vague and general" to 5 = "Precise and specific"
- Overall Quality: 1 = "Very low" to 5 = "Very high"

3. **Question Set Evaluation** (for anonymized Set A and Set B):
   - Gap Targeting: Select which knowledge gaps each sub-question addresses (multi-select including "Does not directly target identified gaps")
   - Precision Rating (4-point scale): 1 = "Broad" to 4 = "Precise"
4. **Simulated Conversation Review**: Evaluators read the simulated tutoring conversation using each question set, then selected which gaps were fully resolved.
5. **Conversation Authenticity** (5-point scale): "Please evaluate whether the given conversation appropriately simulates a realistic student-tutor tutoring situation." 1 = "Very inappropriate" to 5 = "Very appropriate"
6. **Overall Comparison**: "Which question set is better overall?" (Set A / Set B / Both similar)

Method assignment to Set A/B was randomized across conversations. No risks to participants were identified; the task involved reviewing educational content only.

C.4.2 Descriptive Statistics: Generated Content Quality

We first report the absolute quality of generated content as rated by human experts.

Knowledge Gap Quality (5-point scale). Expert raters evaluated the quality of LLM-identified knowledge gaps across three dimensions:

Table 8: Knowledge Gap Quality Ratings (5-point scale: 1=Poor, 5=Excellent). Pooled across all valid raters and datasets.

Dimension	Mean	SD	Median	Range
Comprehensiveness	4.67	0.67	5.0	2–5
Precision	4.45	0.84	5.0	2–5
Overall Quality	4.45	0.74	5.0	2–5

All three dimensions show high ratings (M $\approx$ 4.5/5, Median = 5.0), indicating that LLM-identified knowledge gaps are comprehensive, precise, and of high overall quality according to expert educators.

Generated Question Quality. We analyzed the quality of generated sub-questions across two aspects:

1. Gap Targeting Precision (4-point scale): Raters evaluated how precisely each sub-question targets the identified knowledge gap.

Table 9: Sub-Question Precision Ratings (4-point scale: 1=Poor, 4=Excellent). Analyzed separately for CogKC and LLMKT generated questions.

Method	Mean	SD	Median	Range	N
CogKC	2.79	1.14	3.0	1–4	234
LLMKT	2.69	1.17	3.0	1–4	252
Pooled	2.73	1.16	3.0	1–4	486

Generated questions show good precision (M = 2.73/4, Median = 3.0), with both methods achieving similar quality ratings.

2. Gap Targeting Behavior: Raters annotated which knowledge gaps each sub-question targets, allowing us to quantify targeting specificity.

Table 10: Number of Knowledge Gaps Targeted per Sub-Question. Analysis of gap targeting specificity.

Method	Mean	SD	Median	$\geq$1 Gap	N
CogKC	0.78	0.61	1.0	68.5%	352
LLMKT	0.76	0.61	1.0	66.9%	387
Pooled	0.77	0.61	1.0	67.7%	739

Questions target an average of 0.77 gaps per sub-question, with 67.7% directly targeting at least one identified gap. This indicates focused, targeted question design rather than overly broad questions.

C.4.3 Inter-Rater Reliability

We calculated inter-rater reliability on the 21 shared conversations (3 evaluation sets with 7 conversations each) using weighted Cohen's κ for ordinal precision scores and standard Cohen's κ for binary gap identification:

- Gap Targeting Precision: κ_w = 0.335 (fair to moderate agreement)
- Gap Identification: κ = 0.454 (moderate agreement)
- Better-Set Judgment: Agreement rate = 47.6% (10/21 conversations where both raters chose same winner, excluding ties)

The moderate agreement levels ($\kappa \approx$ 0.3-0.5) indicate that question quality evaluation is inherently subjective, making LLM judge validation particularly important for establishing reliability.

C.4.4 LLM-as-Judge Validation Results

Table 11 presents LLM-human agreement across all three evaluation dimensions on 21 shared conversations.

LLM-human agreement matches or exceeds inter-human reliability across all dimensions. For precision scoring, $\kappa_w = 0.334$ (LLM-human) equals the inter-human baseline ($\kappa_w = 0.335$); consensus comparison reaches $\kappa_w = 0.385$. Gap identification achieves Jaccard similarity of 0.490, 84.5% of the inter-human baseline (0.580). Better-Set judgments show 45.2% agreement with individual raters and 50.0% with consensus, matching the 47.6% inter-human rate. These results validate our LLM-as-judge approach for RQ2.

C.5 Representative Cases of Personalized Questions

This subsection presents a representative example from our empirical evaluation, illustrating how the evaluation methodology works: both methods generate questions independently based on their own KC assessments, then a standardized gap identification process enables fair comparison of question quality.

C.5.1 Example 1: Group Size Synthesis Gap (Conversation 0158_5000241)

Problem

Dani brings 2.5 dozen cupcakes (30 total) for her class of 27 students, 1 teacher, and 1 aid. If 3 students are absent, how many cupcakes are left after giving one to everyone? *Answer:* 4 cupcakes

Conversation

> **Teacher:** Hi Jordy, please tell me about how you got your answer of 6.
>
> **Student:** I started by figuring out that two and a half dozen cupcakes is equal to 30 cupcakes. Then I counted the people in the class, which was 27 + 1 + 1 = 29. After 3 students called in sick, there were only **27 - 3 = 24 students present**. So, I figured Dani needed 24 cupcakes for each student [...] So, I subtracted 24 from 30 and got 6 cupcakes left.
>
> **Teacher:** Let's look again at the last part of the question. It says that "everyone" in the class gets a cupcake. Does that include the teacher and aid?
>
> **Student:** Yes, it does include the teacher and aid, so I added two more people to the equation, making it 27 + 1 + 1 = 29 people in the class. After 3 students called in sick, there were only **27 - 3 = 24 students present**. So, I figured Dani needed 24 cupcakes for everyone in the class [...]

Student Error

The student correctly identifies 29 total people but repeatedly subtracts absences only from students (27-3=24), failing to synthesize the final group size (29-3=26). Answered 6 cupcakes instead of 4.

Phase 1: Independent Question Generation Both methods follow their three-stage pipelines (KC extraction → mastery assessment → question generation) independently.

LLMKT Pipeline Results:

- **Stage 1 - KC Extraction**: 16 flat KCs extracted
- **Stage 2 - Mastery Assessment**: Identified 2 NON_MASTERY KCs:
  - LLMKT_4: "Identifying all recipients in distribution problems"
  - LLMKT_9: "Reading comprehension of problem statements"
- **Stage 3 - Question Generation**: Generated 2 questions targeting these NON_MASTERY KCs:
  (a) "Before considering any absences or field trips, how many people in total was Mr. Harrison planning to give a sticker to?"
  (b) "The problem mentions '25 students (including one new student who just joined)'. How many students are actually in Mr. Harrison's class according to this statement?"

CogKC Pipeline Results:

- **Stage 1 - KC Extraction**: 16 hierarchical KCs extracted (11 Layer-0 + 5 Layer-1)
- **Stage 2 - Mastery Assessment**: Identified 2 NON_MASTERY KCs:
  - KC_L0_S6_004 (ASSESS - NON_MASTERY): "Solution validity" — Think-aloud: "To VALIDATE my answer, I'll re-read the problem"
  - KC_L1_S6_023 (ASSESS - NON_MASTERY): "Problem interpretation correctness" — Think-aloud: "Total cupcakes: 2.5 dozen = 30. Correct."
- **Stage 3 - Question Generation**: Generated 1 question targeting these NON_MASTERY KCs (prioritizing the more specific Layer 1):

Metric	LLM-Human	Consensus	Inter-Human
Gap Targeting Precision (Sub-question Level)			
Weighted Kappa	$\kappa_w = 0.334$	$\kappa_w = 0.385$	$\kappa_w = 0.335$
Gap Identification (Sub-question Level)			
Jaccard Similarity	0.490	0.526	0.580
Exact Match Rate	38.3%	45.6%	52.3%
Better-Set Judgment (Conversation Level)			
Agreement Rate	45.2%	50.0%	47.6%

Table 11: LLM Judge Validation Results on 21 shared conversations. LLM-Human: LLM vs. each human rater. Consensus: LLM vs. human consensus (agreement between both raters). Inter-Human: Agreement between human raters.

(a) "A student, Alex, tried to solve this problem and came up with the following steps: [shows worked solution with error]. Review Alex's work. Do you agree with all of Alex's steps and final answer? If not, identify any errors and explain how to correct them."

Phase 2: Standardized Gap Identification (For Evaluation) To ensure fair evaluation, we perform **standardized gap identification** separately from each method's KC-based assessment. An LLM analyzes the original student-tutor conversation to identify specific knowledge gaps,[4] which serve as ground truth for evaluating both methods' questions.

- **gap_1** (Primary): "Failure to synthesize multiple pieces of information to correctly determine the final group size for distribution. The student correctly identifies the initial total number of people (27 students + 1 teacher + 1 aid = 29) and the number of students who are sick (3). However, they fail to combine these facts to calculate the *actual total number of people present* (24 students + 1 teacher + 1 aid = 26). Instead, they incorrectly use only the number of students present (24) as the total group size for cupcake distribution, effectively omitting the teacher and aid from the final count, despite acknowledging their inclusion in 'everyone' after prompting. This indicates a difficulty in integrating changes to a subset into the overall group calculation."

Phase 3: Evaluation Against Standardized Gap The LLM judge evaluates how well each method's questions target the standardized gap (gap_1), using pairwise comparison with position swapping to prevent bias.

Method	Questions	Target?	Prec.
LLMKT	(a) Calc. total (b) Interpret count	No, No	1.0
CogKC	(a) Review solution for errors	Yes	4.0

Table 12: Question evaluation against standardized gap (gap_1).

- LLMKT Question (a): Asks for initial total people, which the student correctly calculated (29 people). Does not address gap_1 (synthesizing the final count after absences). Precision: 1.0 (Unfocused).
- LLMKT Question (b): Focuses on interpreting student count phrasing, which the student correctly identified. Does not target gap_1. Precision: 1.0 (Unfocused).
- CogKC Question (a): By asking the student to review Alex's worked solution, this question forces them to analyze the step where absences should be subtracted from the total group (not just students), directly targeting gap_1. Precision: 4.0 (Surgical).

Method	#Q	Precision	Coverage	Winner
LLMKT	2	1.0	0/1	
CogKC	1	4.0	1/1	✓

Table 13: Evaluation results for Example 1.

Analysis This example illustrates the key difference between the two approaches. LLMKT identified "identifying all recipients" and "reading comprehension" as gaps, but the student had already

[4]The gap identification prompt also outputs category labels (primary/secondary/tertiary) and competency domains, but these were not used in our analysis as 79% of gaps were classified as primary, making the distinction uninformative for comparative evaluation.

demonstrated these competencies—they correctly identified all groups (27+1+1=29) and read the problem properly. The actual gap was synthesizing the final count after absences. CogKC's hierarchical representation, with explicit cognitive operations (ASSESS) and granularity levels, identified that the student's gap involved assessing whether their final group calculation correctly synthesized all information. This led to a question that directly surfaced the synthesis failure through error analysis.

C.5.2 Example 2: Proportional Language Interpretation (Conversation 0326_5000502)

Problem

Louise is hanging 30 of her pictures on the wall. She hangs some of them vertically, half of them horizontally, then hangs the remaining 5 pictures haphazardly. How many pictures did Louise hang vertically? *Answer:* 10 pictures

Conversation

> **Student:** I divided 15 by 2 to get 7.5, since Louise was hanging **half of the pictures vertically**. Since she couldn't hang half a picture, I concluded that she must have hung 7 pictures vertically.
>
> **Teacher:** Check again whether she hangs half of the pictures vertically or not?
>
> **Student:** Yes, I did. I divided 15 by 2 to get 7.5, since Louise was hanging **half of the pictures vertically**. [...]
>
> **Teacher:** 10 is the required answer. The last step is unnecessary and incorrect because **it is nowhere written that Louise was hanging half the pictures vertically**.

Student Error

The student persistently misinterprets "half of them horizontally" as implying "half of them vertically," even after explicit teacher correction.

Phase 1: Independent Question Generation

LLMKT Pipeline Results:

- **Stage 1 - KC Extraction**: 16 flat KCs extracted
- **Stage 2 - Mastery Assessment**: Identified 1 NON_MASTERY KC:
  - LLMKT_1: "Reading Problem Text" — The ability to carefully read and understand the narrative of a word problem, processing the language to grasp the scenario and context before extracting mathematical information.
- **Stage 3 - Question Generation**: Generated 1 question targeting this NON_MASTERY KC:
  (a) "According to the problem, what is the total number of books Maria is organizing?"

CogKC Pipeline Results:

- **Stage 1 - KC Extraction**: 24 hierarchical KCs extracted (13 Layer-0 + 11 Layer-1)
- **Stage 2 - Mastery Assessment**: Identified 2 NON_MASTERY KCs with prerequisite context:
  - KC_L0_S7_001 (ASSESS - NON_MASTERY): "Solution verification" — Think-aloud: "Finally, I should ASSESS my solution"
    ↪ requires: KC_L0_S5_013 (CALCULATE - MASTERY): "Addition" — "10 + 15 + 5 = 30"
  - KC_L1_S4_015 (IDENTIFY - NON_MASTERY): "Pronoun referents" — Think-aloud: "'Them' refers to the 30 pictures"
    ↪ enables: KC_L0_S4_007 (INFER - MASTERY): "Derive implicit info" — "15 pictures horizontally"
- **Stage 3 - Question Generation**: Generated 2 questions targeting these NON_MASTERY KCs:
  (a) "In the phrase 'a quarter of them', what specific number of books does 'them' refer to?" [targets KC_L1_S4_015 - IDENTIFY]
  (b) "Suppose you found that Sarah placed 15 books on the top shelf. Describe the steps you would take to check if this answer is correct and consistent with the information given in the problem." [targets KC_L0_S7_001 - ASSESS]

Phase 2: Standardized Gap Identification

- **gap_1** (Primary): "The student consistently misinterprets the problem statement, specifically inferring that 'half of the pictures' were

hung vertically, despite the text only stating 'half of them horizontally.' This leads to an incorrect calculation and conclusion regarding the number of vertically hung pictures."

- **gap_2** (Primary): "The student demonstrates a lack of self-monitoring and verification of their reasoning against the original problem constraints. Even after the teacher explicitly prompts them to 'check again whether she hangs half of the pictures vertically or not?', the student reiterates the same incorrect assumption without re-evaluating the problem text."

Phase 3: Evaluation Against Standardized Gaps

Method	Questions	Targets?	Prec.
LLMKT	(a) Identify total qty	No, No	1.0
CogKC	(a) "them" referent (b) Verification	Yes, Yes	4.0

Table 14: Two-gap example: CogKC addresses both gaps while LLMKT addresses neither.

- **LLMKT Question (a)**: Asks for total quantity, which the student correctly identified. Does not target either gap_1 (misinterpreting proportional language) or gap_2 (lack of verification). Precision: 1.0 (Unfocused).
- **CogKC Question (a)**: Directly targets gap_1 by forcing the student to identify what "them" refers to in a proportional phrase, addressing the root cause of their misinterpretation. Precision: 4.0 (Surgical).
- **CogKC Question (b)**: Directly targets gap_2 by prompting metacognitive reflection on verification processes, addressing the student's failure to re-evaluate assumptions against problem constraints. Precision: 4.0 (Surgical).

Method	#Q	Precision	Coverage	Winner
LLMKT	1	1.0	0/2	
CogKC	2	4.0	2/2	✓

Table 15: Multiple-gap example: CogKC achieves complete gap coverage with surgical precision.

Analysis This example demonstrates CogKC's advantage with **multiple gaps**: CogKC's hierarchical representation with explicit cognitive operations enabled it to identify two distinct NON_MASTERY KCs that precisely aligned with both standardized gaps. KC_L1_S4_015 (IDENTIFY operation) targets gap_1 (misinterpreting pronoun referents), while KC_L0_S7_001 (ASSESS operation) targets gap_2 (lack of verification). In contrast, LLMKT identified only "Reading Problem Text"—a generic skill the student had already demonstrated, missing both the specific interpretation failure and the metacognitive verification gap entirely.

C.5.3 Example 3: Solution Verification Gap (Conversation 0219_5000439)

Problem

Jennifer will be 30 years old in ten years. At that time, her sister Jordana will be three times as old as Jennifer. How old is Jennifer's sister now? *Answer:* 80 years

Conversation

> **Teacher:** Does your answer check out with the given conditions?
>
> **Student:** Yes, my answer checks out with the given conditions. Jennifer is currently 20 years old and her sister Jordana is currently **110 years old, which is three times as old as Jennifer in ten years**.
>
> **Teacher:** Not quite. When Jennifer is 30 in 10 years, how old is her sister?
>
> **Student:** When Jennifer is 30 in 10 years, her sister Jordana will be 3 times as old as Jennifer, which means **Jordana will be 90 years old**.

Student Error

The student claims Jordana is "currently 110" but also "will be 90 in 10 years"—a direct logical contradiction they fail to notice without teacher guidance.

Phase 1: Independent Question Generation

LLMKT Pipeline Results:

- **Stage 1 - KC Extraction**: 17 flat KCs extracted
- **Stage 2 - Mastery Assessment**: Identified 1 NON_MASTERY KC:
  - LLMKT_16: "Basic Arithmetic Accuracy" — The overall precision and

correctness in performing fundamental mathematical operations such as addition, subtraction, and multiplication. This encompasses not just knowing how to do the operations but doing them without computational errors.

- **Stage 3 - Question Generation**: Generated 1 question targeting this NON_MASTERY KC:
  (a) "Based on the problem, what is the result of multiplying Sarah's future age by 2?"

CogKC Pipeline Results:

- **Stage 1 - KC Extraction**: 22 hierarchical KCs extracted (12 Layer-0 + 10 Layer-1)
- **Stage 2 - Mastery Assessment**: Identified 2 NON_MASTERY KCs with prerequisite context:
  - KC_L1_S6_026 (ASSESS - NON_MASTERY): "Consistency checking" — Think-aloud: "Is 90 three times 30? Yes, 3 * 30 = 90"
    ↪ requires: KC_L1_S6_025 (CALCULATE - MASTERY): "Deriving values" — "If Jordana is 80 now, in ten years she will be 90"
  - KC_L1_S6_022 (ASSESS - NON_MASTERY): "Solution validation" — Think-aloud: "Let me ASSESS my solution"
    ↪ requires: KC_L0_S4_004 (CALCULATE - MASTERY): "Multiplication" — "3 * 30 = 90"
- **Stage 3 - Question Generation**: Generated 1 question targeting both NON_MASTERY KCs:
  (a) "Suppose you calculated David's current age to be 75 years old. Explain how you would check if this answer is consistent with all the information given in the original problem. Show your work for this check, demonstrating how you would verify both Mark's and David's ages in five years." [targets both KC_L1_S6_026 and KC_L1_S6_022 - ASSESS]

Phase 2: Standardized Gap Identification

- **gap_1** (Primary): "The student demonstrates a significant gap in verifying mathematical solutions and their reasonableness. They failed to identify a direct logical contradiction in their own statement, claiming that Jordana's current age (110 years) was equivalent to Jordana's future age (90 years, derived from the problem's condition of being three times Jennifer's future age). This indicates a lack of critical self-assessment of their proposed solution against the problem's explicit conditions."

Phase 3: Evaluation Against Standardized Gap

Method	Questions	Target?	Prec.
LLMKT	(a) Calculate multiplication	No	1.0
CogKC	(a) Explain verification	Yes	4.0

Table 16: Verification gap: CogKC targets metacognitive verification while LLMKT targets computation.

- **LLMKT Question (a)**: Tests computational skill (multiplying future age) that the student already demonstrated correctly. Does not address gap_1 (verification failure). Precision: 1.0 (Unfocused).
- **CogKC Question (a)**: Surgically targets gap_1 by explicitly prompting the student to describe and demonstrate a verification process, forcing them to check proposed solutions against problem conditions and identify logical contradictions. Precision: 4.0 (Surgical).

Method	#Q	Precision	Coverage	Winner
LLMKT	1	1.0	0/1	
CogKC	1	4.0	1/1	✓

Table 17: Metacognitive gap example: CogKC identifies higher-order reasoning failure.

Analysis This example demonstrates **complete misalignment** between LLMKT's gap characterization and the actual student error. The student's error was a logical contradiction (claiming Jordana is

110 now but 90 in 10 years)—a metacognitive verification failure, not arithmetic. LLMKT misdiagnosed this as "Basic Arithmetic Accuracy," testing multiplication the student could already perform correctly. In contrast, CogKC's two ASSESS operations (KC_L1_S6_026 for consistency checking, KC_L1_S6_022 for solution validation) precisely targeted the verification gap. This misalignment has downstream consequences: LLMKT's question completely fails to address the gap (0% resolution in RQ3 tutoring), while CogKC's verification question achieves full resolution (100%).

Limitations: Cases Where CogKC Underperforms. While the previous examples demonstrate CogKC's advantages, we present two cases where LLMKT outperforms CogKC to provide a balanced analysis. These cases reveal important trade-offs in our approach.

C.5.4 Example 4: Precision vs. Coverage Trade-off (Conversation 0364_5001026)

Problem

Noah calls his Grammy every week for 30 minutes at $0.05 per minute. How much would he be billed for a year? *Answer:* $78

Conversation

> **Teacher:** How many minutes does each call last?
>
> **Student:** Each call lasts 30 minutes.
>
> **Teacher:** So why have you multiplied that by 2?
>
> **Student:** Oh, I see. I multiplied 30 minutes by 2 because I was calculating the total number of minutes for the year, which would be **2 x 30 minutes = 60 minutes for each call**.

Student Error

The student made two distinct errors: (1) misreading/misapplying the 30-minute duration, and (2) confusing single-event duration with total aggregation over time.

Phase 1: Independent Question Generation

LLMKT Pipeline Results:

- **Stage 1 - KC Extraction**: 24 flat KCs extracted
- **Stage 2 - Mastery Assessment**: Identified 6 NON_MASTERY KCs:
  - LLMKT_13: "Reading for Detail"
  - LLMKT_24: "Attention to Problem Details"
  - LLMKT_1: "Problem Comprehension"
  - LLMKT_2: "Identify Given Information"
  - LLMKT_11: "Contextual Interpretation"
  - LLMKT_18: "Understanding Time Units"
- **Stage 3 - Question Generation**: Generated 6 questions targeting these NON_MASTERY KCs

CogKC Pipeline Results:

- **Stage 1 - KC Extraction**: 27 hierarchical KCs extracted (14 Layer-0 + 13 Layer-1)
- **Stage 2 - Mastery Assessment**: Identified 1 NON_MASTERY KC:
  - KC_L1_S1_003 (EXTRACT - NON_MASTERY): "All relevant numerical and quantitative information from a problem statement"
- **Stage 3 - Question Generation**: Generated 1 question targeting this NON_MASTERY KC

Phase 2: Standardized Gap Identification

- **gap_1** (Primary): Misinterpreted the call duration, using 60 minutes instead of 30 minutes despite the problem clearly stating "30 minutes."
- **gap_2** (Primary): Confusion between single-event duration and aggregation over multiple events—incorrectly justified multiplying by 2 as calculating "total minutes for the year."
- **gap_3** (Secondary): Failed to monitor reasoning consistency—did not notice the contradiction of using 30, 60, and "1 hour" interchangeably.

Phase 3: Evaluation Results

Method	#Q	Precision	Coverage	Winner
LLMKT	6	2.75	3/3	✓
CogKC	1	4.0	1/3	

Table 18: Precision vs. Coverage: CogKC achieves surgical precision but incomplete gap coverage.

- **CogKC**: Generated 1 question with perfect precision (4.0) targeting gap_1 (information extraction). However, CogKC's mastery assessment identified only KC_3 ("extract numerical information") as NON_MASTERY, missing the KCs related to gap_2 (aggregation confusion) and gap_3 (reasoning monitoring).

- **LLMKT**: Generated 6 questions with variable precision (range 1.0–4.0, mean 2.75). Questions (a), (d) targeted gap_1; question (f) targeted gap_2; questions (b), (c), (e) were broader scaffolding. Despite lower mean precision, achieved complete gap coverage.

Analysis CogKC achieved surgical precision (4.0) but covered only 1 of 3 gaps, while LLMKT's broader approach (6 questions, mean 2.75) achieved complete coverage. This reflects a precision-coverage trade-off: CogKC's mastery assessment identified only information extraction as NON_MASTERY, missing the aggregation-related gaps. For students with multiple co-occurring gaps, broader coverage may be preferable; CogKC excels when gaps are singular and well-defined.

C.5.5 Example 5: Process-oriented Questions vs. Outcome-level Gaps (Conversation 1817_5000313)

Problem

Alice walks 10 miles to school (morning) and 12 miles home (afternoon) every Monday–Friday. How many miles will she have walked when she gets home on Friday? *Answer:* 110 miles

Conversation

> **Teacher:** Can you please tell me how much Alice walks on Monday?
>
> **Student:** On Monday, Alice walks 10 miles through a large grass field to get to school. So she walks a total of **10 miles on Monday**.
>
> **Teacher:** No. She walks 10 + 12 = 22 miles on Monday.
>
> **Student:** Oh, I see. Yes, Alice walks 10 miles through the grass field to get to school and 12 miles through the forest to get home on Monday, so she walks a total of **10 + 12 = 22 miles on Monday**.

Student Error

The student failed to recognize that *daily* distance includes both to-school and return-home journeys, answering 98 miles instead of 110.

Phase 1: Independent Question Generation

LLMKT Pipeline Results:

- **Stage 1 - KC Extraction**: 26 flat KCs extracted
- **Stage 2 - Mastery Assessment**: Identified 9 NON_MASTERY KCs:
  - LLMKT_12: "Identifying Number of Days"
  - LLMKT_23: "Interpreting That Week"
  - LLMKT_26: "Verifying the Question"
  - LLMKT_5: "Understanding Time Periods"
  - LLMKT_6: "Interpreting Daily Routine"
  - LLMKT_13: "Calculating Daily Total Distance"
  - LLMKT_14: "Applying Addition for Daily Travel"
  - (+ 2 more)
- **Stage 3 - Question Generation**: Generated 7 questions (maximum limit)

CogKC Pipeline Results:

- **Stage 1 - KC Extraction**: 31 hierarchical KCs extracted (14 Layer-0 + 17 Layer-1)
- **Stage 2 - Mastery Assessment**: Identified 6 NON_MASTERY KCs:
  - KC_L1_S2_005 (IDENTIFY): "An initial, simpler sub-problem within a complex problem"
  - KC_L1_S1_003 (INFER): "The overall temporal scope or conditions for a problem's calculation"
  - KC_L0_S1_011 (INFER): "The meaning, goal, scope, and specific requirements of a problem's core question"
  - KC_L1_S7_036 (VALIDATE): "The correctness and validity of the problem-solving process"
  - (+ 2 more validation/assessment KCs)
- **Stage 3 - Question Generation**: Generated 6 questions targeting these NON_MASTERY KCs

Phase 2: Standardized Gap Identification

- **gap_1** (Primary): Failed to recognize that a single day's total distance includes both journeys (morning + afternoon).
- **gap_2** (Primary): Misunderstood the temporal scope "when she gets home on Friday," excluding Friday's return journey.
- **gap_3** (Primary): Failed to consistently apply corrected understanding in subsequent calculations.
- **gap_4** (Primary): Double-counted partial and full journey components when correcting.

Phase 3: Evaluation Results

Method	#Q	Precision	Coverage	Winner
LLMKT	7	3.5	4/4	✓
CogKC	6	2.9	3/4	

Table 19: Process vs. outcome-level: CogKC's process-oriented questions receive lower precision scores against outcome-level gaps.

- **LLMKT**: Questions directly targeted the specific conceptual gaps:
  (e) "For a single day, how many separate driving segments does Mark complete?" → gap_1 (Precision: 4.0)
  (f) "What is the total distance on one single day? Show calculation." → gap_1 (Precision: 4.0)
  (b) "What specific period does 'that week' refer to?" → gap_2 (Precision: 4.0)
- **CogKC**: Questions focused on higher-level metacognitive skills:
  (a) "What is the most logical initial sub-problem?" → Problem decomposition (Precision: 4.0)
  (d) "How would you validate your problem-solving process?" → Metacognitive (Precision: 2.0)
  (f) "How would you assess consistency?" → Metacognitive (Precision: 2.0)

Analysis CogKC's questions target cognitive processes ("What initial sub-problem should you solve?") rather than direct outcomes ("How many segments per day?"). When gaps are defined at the outcome level, process-oriented questions receive lower precision scores even when they address the same underlying skills. However, RQ3 demonstrates that CogKC's process-oriented approach leads to higher gap resolution rates in tutoring simulations, validating cognitive-level interventions despite lower precision scores in this evaluation framework.

C.6 Statistical Results for Question Evaluation

Table 20 provides complete statistical details for all pairwise comparisons reported in the main text, including LLM winner rates and raw precision and coverage scores.

C.7 Baseline Variant Analysis

To test whether CogKC's advantages stem from hierarchical structure rather than mere information quantity, we evaluated against two conditions for each baseline method: BASIC (controlled for same KC count as CogKC) and FULL (same KC count as CogKC, plus matched information amount via character-based constraint). Table 21 presents win rates across both conditions. CogKC (ML=1) maintains consistent advantages, demonstrating that hierarchical multi-faceted structure provides benefits beyond mere information quantity.

Interpretation Controlling for information volume reveals the source of CogKC's advantage. FULL variants—matching both KC count and character length—substantially strengthen baselines: LLMKT-FULL closes the gap to 56.3% versus LLMKT-BASIC's 59.2%, and KCQRL-FULL to 60.6% versus 64.5%. Yet CogKC still wins consistently. If information quantity alone explained our results, these advantages should disappear when baselines match CogKC's volume; they do not.

The advantage gap varies by baseline. Against KCQRL (60.6–64.5%), CogKC outperforms more decisively than against LLMKT (56.3–59.2%), likely reflecting differences in how each method extracts KCs. The largest gap appears against Vanilla (67.1%), confirming that any structured KC representation—hierarchical or flat—improves over real-time gap analysis.

Table 20: Question Generation Performance Comparison (RQ2). Evaluated on N=2,861 conversations with pairwise LLM-as-judge comparison. LLM Winner Rate shows percentage of pairwise wins. Precision: mean gap targeting precision score across all sub-questions in a set (1-4 scale, averaged per conversation). Gap Coverage: percentage of identified gaps addressed (0-100% scale, per conversation).

Comparison	LLM Winner Rate (%)	Precision (1-4)		Gap Coverage (0-100%)	
		Method A	Method B	Method A	Method B
vs LLMKT-FULL (baseline)					
ML=1	**56.3*** **	3.18 ± 0.72	2.99 ± 0.73	87.8 ± 24.4	85.4 ± 26.3
ML=1-CO	50.0	3.20 ± 0.52	$3.21 \pm 0.50^{\ddagger}$	88.1 ± 23.2	86.2 ± 23.8
ML=0	$44.1^{\dagger}$	3.24 ± 0.56	3.21 ± 0.52	84.6 ± 25.2	85.9 ± 24.3
ML=1 vs other baselines					
vs KCQRL-FULL	**60.6*** **	3.31 ± 0.49	3.09 ± 0.53	88.7 ± 22.2	85.4 ± 25.3
vs Vanilla	**67.1*** **	3.35 ± 0.50	3.05 ± 0.56	89.4 ± 21.8	83.4 ± 25.1
Ablation studies					
ML=1 vs ML=1-CO	**56.7*** **	3.21 ± 0.70	3.03 ± 0.72	88.5 ± 23.4	87.8 ± 24.6
ML=1 vs ML=0	**61.4*** **	3.28 ± 0.49	3.20 ± 0.55	88.7 ± 22.3	83.8 ± 25.6

†ML=0 loses to LLMKT-FULL (LLMKT wins 53.9%). ‡p = 0.255 (not significant).
***$p < 0.001$ (paired t-test, N=2,861 conversations). All other comparisons significant.

Table 21: Question Generation Win Rates: Basic vs Full Variants (RQ2). Pairwise LLM-as-judge evaluation on N=2,861 conversations. Basic variants: same KC count as CogKC. Full variants: same KC count + matched information amount (character-based). ***$p < 0.001$ (paired t-test).

Baseline	Variant	ML=1 (%)	Baseline (%)	Tie (%)
LLMKT	Basic	**59.2*** **	39.3	1.4
	Full	**56.3*** **	42.5	1.2
KCQRL	Basic	**64.5*** **	33.8	1.7
	Full	**60.6*** **	37.7	1.7
Vanilla	—	**67.1*** **	31.7	1.2

C.8 Question Generation Pipeline

This section provides detailed prompts and examples illustrating the complete workflow for both KC-based and Vanilla question generation approaches.

C.8.1 KC-Based Generation: Stage 1 - KC Extraction

Prompt Template (Simplified)

```
Extract knowledge components from this problem-solving process.
Each KC should specify:
- Cognitive Action (what mental operation)
- Content Element (what math concept/procedure)
- Learning Outcome (what student should achieve)
```

Input: Original problem + Think-aloud protocol

```
Problem: Julia bought a package of spoons. Her husband
bought 5 more. She used 3 to sample stew. She had 12
total. How many spoons were in Julia's package?

Think-aloud Protocol (excerpt):
"First, IDENTIFY what problem asks - find initial qty.

EXTRACT numerical information:
1. Julia bought a package (unknown)
2. Husband bought 5 spoons
3. Julia used 3 spoons
4. Final count: 12 spoons

DECOMPOSE this problem:
Part 1: Sequence (buying, receiving, using)
Part 2: What does 12 represent
Part 3: Work backward to find initial purchase

Part 1: Julia bought x, received 5, used 3, has 12.
Part 2: '12' includes used ones. 12 + 3 = 15 total.
Part 3: Total=15, husband gave 5. Julia: 15-5=10"

(Protocol continues ~2000 words)
```

Output: Extracted KCs

```
KC_L0_S1_001:
  Action: calculate
  Content: inverse operations
  Outcome: apply addition to reverse subtraction

KC_L0_S2_001:
```

Table 22: Example KC representations across baseline variants for the same mathematical skill. CogKC provides hierarchical structure with multi-faceted representation, while baselines use flat lists. FULL variants extend text to match CogKC's information volume but maintain flat structure.

Method	KC Representation

CogKC	`[KC_19] (PARENT)`
	cognitive_action: execute
	content_element: Proportional calculations for sequential changes
	learning_outcome: Execute multi-step proportional calculations
	think_aloud: "So, if there was 1/2 of the water, and half of *that* spills..."
	`|- [KC_20] (SUB-FULLONENT)`
	cognitive_action: calculate
	content_element: Product of two fractions
	learning_outcome: Calculate product of two given fractions
	think_aloud: "This equals 1/4."

LLMKT-BASIC	`name:` Multiply two fractions `description:` Perform multiplication of two fractional values.
LLMKT-FULL	`name:` Multiply Fractions `description:` This operation involves multiplying two fractional values by multiplying numerators together and denominators together. Students must understand that this represents finding a fraction of a fraction and apply the formula correctly.
KCQRL-BASIC	`label:` Multiplication of fractions
KCQRL-FULL	`label:` Understanding Fractions – Fractions represent parts of a whole or collection. They consist of a numerator and denominator. Concepts like "half" (1/2) are central to describing portions, requiring clear grasp of their meaning.

```
Action: synthesize
Content: multi-step tracking
Outcome: combine operations to find unknowns
```

C.8.2 KC-Based Generation: Stage 2 - KC Mastery Assessment

Prompt Template (Simplified)

```
Analyze student's conversation to assess mastery of each KC.
Classify as: MASTERY / NON_MASTERY / INSUFFICIENT_INFO
Provide confidence score (0-1) and evidence from conversation.
```

Input: Extracted KCs + Student-tutor conversation

```
(See extracted KCs from Stage 1 above
 + original student-tutor conversation)
```

Output: KC Mastery Assessment

```
KC_L0_S1_001 (inverse operations):
- Mastery State: NON_MASTERY
- Confidence: 0.85
- Evidence: Student computed 12-3=9 (should be 12+3)
- Reasoning: Wrong operation to reverse "used" qty

KC_L0_S2_001 (multi-step quantity tracking):
- Mastery State: MASTERY
- Confidence: 0.90
- Evidence: Tracked: 12+3=15, then 15-5=10
- Reasoning: Combined sequential operations correctly
```

C.8.3 KC-Based Generation: Stage 3 - Question Generation

Prompt Template (Simplified)

```
Generate ONE problem with sub-questions for NON_MASTERY KCs.

Main Problem Guidelines:
- Similar to original (same scenario type)
- Change surface details (numbers, names)

Sub-Question Guidelines:
- Each targets EXACTLY ONE KC
- Label as (a), (b), (c), etc.
- Make questions independent
```

Input: NON_MASTERY KCs + Original problem context

```
KC_L0_S1_001 (NON_MASTERY):
  inverse operations for working backward

+ Original problem context from Stage 1
```

Output: Generated Follow-up Questions

```
Main Problem: Maria baked cookies. Sarah brought 8 more.
Her brother ate 5. She had 25 ready. How many did Maria
bake?

Sub-questions:
(a) Sarah brought 8 more. If 'x' is what Maria baked,
    write expression for total *after* Sarah, *before*
    brother. [Targets: KC_L0_S2_001 - multi-step]

(b) Maria had 25 *after* brother ate 5. To find *before*
    what operation? [Targets: KC_L0_S1_001 - inverse ops]
```

C.8.4 Vanilla Generation: Single-Stage Workflow

Prompt Template Reference

```
See Appendix (Vanilla Mode section) for complete prompt.
```

Input: Original problem + Student-tutor conversation

```
(Same Julia/spoons problem from Stage 1
 + student-tutor conversation)
```

Output: Gap Identification + Question Generation in single LLM call

```
Main Problem: Liam collects comics. Friend gave 7, he
traded 4, had 15 total. How many did Liam start with?

Sub-questions:
(a) What unknown must you find? Use a variable.
    [Targets: identify core question]

(b) If 'x' is comics Liam started with, expression for
    comics after friend gave him some.
    [Targets: 'giving' = addition]

(c) If 'y' is comics before trade, expression for after.
    [Targets: 'trading away' = subtraction]

(d) Write one equation for the entire problem.
    [Targets: synthesizing into equation]

(e) Solve equation from (d). Show steps.
    [Targets: solving linear equations]

(f) How would you check your answer?
    [Targets: verify solutions]
```

Key Difference: Unlike KC-based generation (3 stages: KC extraction → mastery assessment → question generation), Vanilla performs gap identification and question generation in a *single LLM call*, analyzing the conversation directly without decomposing into atomic KCs. Gaps are identified implicitly through each sub-question's rationale rather than through explicit KC extraction. Both approaches generate structurally similar main problems with independent sub-questions.

Note: All examples in this section (Julia/spoons → Maria/cookies for KC-based, Julia/spoons → Liam/comic books for Vanilla) are derived from the same original problem and conversation (conversation ID: 0001_6000025) from our research dataset, demonstrating how each approach transforms the same input into targeted follow-up questions.

D RQ3: Tutoring Effectiveness

This appendix provides supplementary materials for RQ3 (Tutoring Simulation Effectiveness), including complete test statistics, convergence analysis, and robustness testing.

D.1 Simulation Prompts

This section provides the complete prompt specifications for the three components of our tutoring simulation pipeline: student simulator, tutor simulator, and LLM-as-judge evaluator.

D.1.1 Student Simulator Prompt

```
You are a student with specific knowledge gaps
working with a math tutor.
KNOWLEDGE GAPS (concepts you struggle with):
{knowledge_gaps}
CONVERSATION HISTORY: {conversation}
INSTRUCTIONS: - Initially show confusion or
misconceptions in your knowledge gap areas
- Make realistic mistakes related to these
gaps, but learn from good explanations -
Keep responses passive and concise like this
example: "I think the derivative of x² is
2x but I'm not sure why we multiply by the
exponent." - Show your thinking briefly but
don't ask for clarification - When the tutor
provides clear explanations or corrections,
gradually build understanding - After learning
a concept, retain that understanding and
apply it consistently - Don't immediately
revert to the same misconception after being
corrected - Show realistic student learning
progression: confusion → partial understanding
→ application
OUTPUT FORMAT: { "reply": "student_reply" }
```

D.1.2 Tutor Simulator Prompt

```
You are an expert math tutor helping a student
work through a math problem. Your goal is to
provide effective tutoring while guiding the
student toward the correct solution.
TUTORING OBJECTIVES: {follow_up_suggestions}
CONVERSATION HISTORY: {conversation}
TEACHING APPROACH: - Provide direct
explanations when students show confusion
about fundamental concepts - Use brief,
focused responses (under 200 characters)
INSTRUCTIONS: - The main problem and
sub-questions have already been presented in
the conversation history - Start directly
with the first sub-question if this is your
first response after the problem presentation
```

- Label each sub-question clearly (e.g., "Let's start with the first question:", "Now for the second question:") - Do NOT repeat the main problem or list all sub-questions again - Work through sub-questions systematically, one at a time - When ALL sub-questions are completed successfully, end with: "All tutoring objectives completed." - Use conversation history to avoid repeating confirmed understanding - If the student's answer is wrong, provide ONE specific hint - Guide efficiently - no unnecessary steps or elaboration - Stay under 200 characters per response - Focus on mathematical progression, not emotional support
TUTOR RESPONSE (JSON FORMAT ONLY) { "reply": "tutor_reply" }
CHARACTER COUNT CHECK: Before responding, ensure your reply is under 200 characters.

D.1.3 LLM-as-Judge Evaluator Prompt

You are evaluating whether a tutoring session should be terminated based on gap resolution status.
ORIGINAL CONVERSATION (for understanding student's initial state): {original_conversation}
IDENTIFIED KNOWLEDGE GAPS (from original conversation analysis): {knowledge_gaps}
CURRENT TUTORING CONVERSATION TO EVALUATE: {new_conversation}
INSTRUCTIONS: Determine if the tutoring conversation should be terminated by analyzing resolution status of the IDENTIFIED KNOWLEDGE GAPS above.
GAP RESOLUTION EVALUATION: 1. **Check Status of IDENTIFIED KNOWLEDGE GAPS**: For each specific gap listed in IDENTIFIED KNOWLEDGE GAPS above, determine if it has been adequately resolved 2. **Student Understanding**: Look for evidence that student demonstrates comprehension of the concepts from IDENTIFIED KNOWLEDGE GAPS 3. **Learning Progress**: Assess if student can now handle the concepts from IDENTIFIED KNOWLEDGE GAPS independently
TERMINATION CRITERIA:
TERMINATE with "gap_filled" when: - ALL gaps from the IDENTIFIED KNOWLEDGE GAPS list are resolved - Student demonstrates understanding of all concepts from IDENTIFIED KNOWLEDGE GAPS - Student shows ability to apply all concepts from IDENTIFIED KNOWLEDGE GAPS independently
TERMINATE with "explicit_completion" when: - Teacher explicitly states "All tutoring objectives completed" or similar completion phrases - MUST still provide complete gap resolution analysis in gaps_status even for explicit completion
CONTINUE (finished: false) when: - Any gaps from IDENTIFIED KNOWLEDGE GAPS remain unresolved but conversation is progressing - Student is making learning progress on IDENTIFIED KNOWLEDGE GAPS but needs more time - Tutor is successfully guiding student toward resolving IDENTIFIED KNOWLEDGE GAPS
DECISION LOGIC: - Primary focus: Are ALL the IDENTIFIED KNOWLEDGE GAPS resolved? - Secondary focus: Is productive learning still happening on IDENTIFIED KNOWLEDGE GAPS? - Completion signals: Look for phrases like "All tutoring objectives completed." - IMPORTANT: Even when explicit completion phrases are detected, you MUST still analyze the gap resolution status and provide complete gaps_status metadata.
RESPONSE FORMAT (JSON): { "finished": "boolean value", "termination_reason": "gap_filled | explicit_completion | CONTINUE", "reasoning": "detailed explanation of decision based on IDENTIFIED KNOWLEDGE GAPS status", "gaps_status": { "total_gaps_resolved": "number of IDENTIFIED KNOWLEDGE GAPS resolved", "total_gaps": "total number of IDENTIFIED KNOWLEDGE GAPS", "unresolved_gaps": ["list of unresolved gap_ids from IDENTIFIED KNOWLEDGE GAPS"] } }
IMPORTANT: - When finished is false, termination_reason MUST be "CONTINUE" (not null or empty) - When finished is true, termination_reason must be either "gap_filled" or "explicit_completion"

D.2 Evaluation Overview

This appendix provides complete statistical details for the RQ3 tutoring effectiveness evaluation, including all test statistics, standard errors, confidence intervals, effect sizes, and illustrative examples. The appendix is organized as follows: §D.3 presents detailed conversation comparison examples, §D.3.5 provides complete results with all statistics, §D.3.6 interprets statistical significance and effect sizes, §D.3.7 validates sample size through convergence analysis, §D.3.8 examines robustness across turn limits, and §D.3.9 validates LLM judge reliability.

D.3 Representative Cases of Tutoring Conversations

This subsection presents side-by-side comparisons of tutoring conversations generated using CogKC vs. LLMKT baseline methods, illustrating how hierarchical multi-faceted KC representations enable more efficient gap resolution through targeted tutoring interactions.

D.3.1 Example 1: Fraction Operations Misconception

Identified Knowledge Gaps

- Gap 1: Understanding that dividing by a fraction is equivalent to multiplying by its reciprocal
- Gap 2: Correctly applying the reciprocal operation when dividing fractions

- Gap 3: Simplifying fraction results to lowest terms

Tutoring Conversation Comparison

CogKC (ML=1) - 8 turns, All gaps resolved

Turn 1 - Tutor: Let's work on understanding division with fractions. Can you explain what happens when we divide by $\frac{1}{2}$?
Turn 2 - Student: I think it makes the number smaller, like regular division.
Turn 3 - Tutor: Good thinking, but division by fractions works differently! When we divide by $\frac{1}{2}$, we're asking "how many halves fit into our number?" Let's try: how many $\frac{1}{2}$s fit into 1 whole?
Turn 4 - Student: Two halves make a whole, so... 2?
Turn 5 - Tutor: Exactly! So $1 \div \frac{1}{2} = 2$. The result is bigger! Now, can you explain the reciprocal relationship here?
Turn 6 - Student: The reciprocal of $\frac{1}{2}$ is 2, and $1 \times 2 = 2$... So dividing by a fraction is the same as multiplying by its reciprocal?
Turn 7 - Tutor: Perfect! Now let's practice: What is $\frac{3}{4} \div \frac{2}{5}$?
Turn 8 - Student: I multiply $\frac{3}{4} \times \frac{5}{2} = \frac{15}{8}$.
Outcome: All 3 gaps resolved in 8 turns. Efficiency: 0.375 gaps/turn.

LLMKT (Flat KC) - 14 turns, 2/3 gaps resolved

Turn 1 - Tutor: Let's review fraction division. Here's a problem: $\frac{3}{4} \div \frac{2}{5}$. How would you solve this?
Turn 2 - Student: Do I just divide the numerators and denominators?
Turn 3 - Tutor: Not quite. There's a special rule for fraction division. Can you recall what it is?
Turn 4 - Student: Something about flipping?
Turn 5 - Tutor: Yes, we flip the second fraction. So what does $\frac{3}{4} \div \frac{2}{5}$ become?
Turn 6 - Student: $\frac{3}{4} \times \frac{2}{5}$?
Turn 7 - Tutor: Almost! We need to flip $\frac{2}{5}$ to $\frac{5}{2}$. Try again.
Turn 8 - Student: $\frac{3}{4} \times \frac{5}{2} = \frac{15}{8}$.
Turn 9 - Tutor: Good! But why do we flip? What's the mathematical reason?
Turn 10 - Student: I'm not sure, it's just the rule we learned.
Turn 11 - Tutor: Let me explain. Dividing by $\frac{2}{5}$ means "how many $\frac{2}{5}$s fit?" This is equivalent to multiplying by $\frac{5}{2}$. Does that make sense?
Turn 12 - Student: Sort of... I can apply the rule but I'm confused why it works.
Turn 13 - Tutor: Let's try another: $\frac{1}{2} \div \frac{1}{3}$.
Turn 14 - Student: $\frac{1}{2} \times \frac{3}{1} = \frac{3}{2}$.
Outcome: Gaps 2 and 3 resolved, Gap 1 (conceptual understanding) partially addressed. Efficiency: 0.143 gaps/turn.

Analysis CogKC's hierarchical structure enables the tutor to systematically address the conceptual foundation (Gap 1) before procedural application (Gap 2), resulting in deeper understanding and more efficient gap resolution (0.375 vs. 0.143 gaps/turn). The LLMKT approach focuses on procedural execution without establishing the underlying conceptual framework, leading to mechanical application without understanding.

Metric Calculation Details For this example, RQ3 tutoring effectiveness metrics are calculated as follows:

Efficiency (gaps resolved per turn): Number of gaps resolved divided by total conversation turns. We use this ratio to measure how quickly gaps are addressed through tutoring interactions.

- CogKC: 3 gaps resolved / 8 turns = 0.375 gaps/turn
- LLMKT: 2 gaps resolved / 14 turns = 0.143 gaps/turn

Gap Resolution Ratio: Percentage of identified gaps that were successfully resolved by the end of the conversation.

- CogKC: 3 out of 3 gaps resolved = 100%
- LLMKT: 2 out of 3 gaps resolved = 66.7%

Completion Rate: Whether the conversation terminated via *gap_filled* (all gaps resolved) or *explicit_completion* (tutor completed all subquestions), versus *max_turn_reached* (incomplete). Both gap_filled and explicit_completion count as successful completion.

- CogKC: gap_filled (all gaps resolved naturally)
- LLMKT: max_turn_reached (incomplete at turn limit)

Efficiency Win Rate: In head-to-head comparisons across all conversations, percentage where one method achieves higher gaps-per-turn efficiency. This metric captures pairwise performance rather than absolute values.

D.3.2 Example 2: Multi-Step Equation Solving

Identified Knowledge Gaps

- Gap 1: Understanding the order of operations in equation solving
- Gap 2: Correctly isolating variables through inverse operations

Table 23: Tutoring Efficiency Comparison for Multi-Step Equation Example

Method	Turns	Gaps	Eff.	Term.
CogKC (ML=1)	11	2/2 (100%)	0.182	gap
LLMKT-FULL	18	2/2 (100%)	0.111	explicit

Conversation Summary

Key Differences

- **CogKC approach**: Uses grounding examples from ML0 layer to demonstrate order of operations with specific numbers before generalizing to variable expressions. Achieves natural completion when gaps are resolved.
- **LLMKT approach**: Provides abstract procedural rules without concrete grounding, requiring more clarification turns and examples. Continues until all sub-questions are exhausted rather than detecting gap resolution.

D.3.3 Example 3: Group Size Synthesis Gap (Conversation 0158_5000241)

This example demonstrates the same gap from RQ2 (Section C.5), showing how question quality translates to tutoring effectiveness.

Identified Knowledge Gap

- Gap: Failure to synthesize total group size for distribution after accounting for absences (omitting teacher and aid from final count, focusing only on students present)

Tutoring Conversation Comparison

CogKC (1 question) - 20 turns, Gap resolved with deep understanding

Question: Review Alex's worked solution with an error in calculating people present (41 - 6 = 35), identify and correct errors.
Turn 2-3: Student identifies arithmetic error (36+6=42, not 40) ✓
Turn 9: Student: "Alex's Step 4 is wrong. There are 35 students and 5 are absent, so 30 students are present. So there are 30 people present." [Makes synthesis error]
Turn 11: Student: "I see 41 - 6 is 35. But if 5 students are absent, that leaves 30 students. I'm not sure how 35 people are present." [Struggles with synthesis]
Turn 12: Tutor: "How many teachers are present? How many volunteers are present? Add them all up."
Turn 15: Student: "30 + 3 + 2 is 35. So there are 35 people present. But I thought we were only counting the students for the granola bars, so it would be 30." [Breakthrough moment]
Turn 16: Tutor: "The problem says 'everyone who is present.' This includes students, teachers, and volunteers."
Turn 17: Student: "Okay, so 35 people are present, including students, teachers, and volunteers." [Understanding achieved]
Turn 19-20: Student applies corrected understanding to final calculation (42-35=7) ✓
Outcome: Gap resolved, explicit completion with "All tutoring objectives completed."

LLMKT (2 questions) - 13 turns, Gap resolved through scaffolding

Questions: (a) Calculate initial total people. (b) Interpret student count statement.
Turn 3: Student: "25 + 1 + 2 = 28 people in total." [Correctly answers Q1] ✓
Turn 5: Student: "There are 25 students in the class." [Correctly answers Q2] ✓
Turn 7: Student: "25 - 2 - 1 = 22 students were present." ✓
Turn 9: Student: "I think 22 people were present." [Makes THE SAME synthesis error despite answering questions correctly!]
Turn 10: Tutor: "Remember, the teacher and volunteers were also present. How many people in total?"
Turn 11: Student: "22 students + 1 teacher + 2 volunteers = 25. But then 25 - 3 = 22 people." [Still confused]
Turn 12: Tutor: "The 22 students already accounts for those absent. So, how many people were present?"
Turn 13: Student: "22 students + 1 teacher + 2 volunteers = 25 people. The 22 already accounts for absent ones." [Finally resolves through teacher scaffolding]
Outcome: Gap resolved through teacher prompting, not question design. Questions didn't prevent error recurrence.

Analysis This example illustrates **question quality determining tutoring effectiveness**. CogKC's surgical error-analysis question directly surfaces the synthesis failure (turn 9-17), forcing the student to confront and resolve the precise cognitive gap. LLMKT's procedural questions fail to prevent the same synthesis error from recurring (turn 9) despite correct answers—the student makes the identical mistake (counting only students) that caused the original problem. Gap resolution in LLMKT occurs through teacher scaffolding, not question design, demonstrating that **fine-grained KC representation enables questions that preventatively target root causes rather than reactively addressing symptoms**.

D.3.4 Statistical Pattern Across All Examples

Across the full dataset of 1,399 conversations, CogKC demonstrates consistent advantages:

- **Higher Efficiency**: CogKC achieves 14% higher gaps-per-turn efficiency (0.216 vs. 0.189), statistically significant with large effect size (Cohen's $d = 0.52$)
- **Better Gap Resolution**: CogKC resolves more gaps per conversation (79.3% vs. 71.6%), demonstrating more effective targeting
- **More Natural Termination**: CogKC achieves higher rates of natural completion through gap resolution (45.2%) compared to reaching question exhaustion (38.1%), while LLMKT shows opposite pattern (31.8% vs. 52.4%)

These patterns validate that hierarchical multi-faceted KC representations enable more effective tutoring through better conceptual grounding, systematic progression from foundations to applications, and more accurate gap resolution detection.

D.3.5 Statistical Results for Tutoring Effectiveness

Table 24 presents the complete statistical validation across all metrics, with full details on standard deviations, confidence intervals, test statistics, and standard errors.

D.3.6 Statistical Significance and Effect Sizes

CogKC demonstrates statistically significant advantages on two continuous metrics: Efficiency shows the strongest effect ($p < 0.001$, $d = 0.103$, 95% CI [0.007, 0.022]), while Gap Resolution Ratio reaches significance near the conventional threshold ($p = 0.048$, $d = 0.053$, 95% CI [0.0, 5.6]).

Win rates favor CogKC (47.4% vs 39.5% for efficiency, 28.4% vs 23.7% for gap resolution), while completion rates remain nearly identical (76.8% vs 77.1%). This pattern indicates CogKC's advantage lies in efficiency—reaching the same outcomes faster with better gap targeting—rather than raw completion capability.

Small effect sizes ($d < 0.5$) are expected given the constrained context where both methods use identical tutoring approaches with differences arising solely from KC representation quality, and LLMKT's high baseline performance (77% completion) leaves limited room for improvement. Despite small effects, strong statistical significance demonstrates precise measurement, and the 14% efficiency improvement represents meaningful pedagogical gains typical of educational interventions ($d = 0.05$ to 0.20) (Kraft, 2020).

D.3.7 Sample Size Validation

Our evaluation used N=1,399 conversations (599 test set, 800 training set). Since our approach relies entirely on prompting-based methods without model training, traditional train-test separation is unnecessary—all evaluation is performed on previously unseen conversation contexts.

To validate that N=1,399 provides sufficient data for stable conclusions, we employ convergence analysis using bootstrap resampling (1,000 samples). Figure 5 shows the convergence patterns for both efficiency and gap resolution metrics. Two complementary indicators demonstrate stabilization by N=1,100:

Confidence Interval Width: 95% CI widths stabilized below 6% by N=1,100 for all metrics, indicating precise estimates (±3% precision).

Standard Error: Standard errors decreased to $< 1.53\%$ for all metrics by N=1,100, confirming low estimation uncertainty. This threshold corresponds to the 6% CI width target (SE = 6% / 3.92 $\approx 1.53\%$).

The consistent stabilization pattern demonstrates that N=1,399 is sufficient, and additional data collection would not meaningfully change conclusions.

D.3.8 Turn Limit Sensitivity Analysis

We validate robustness of our findings across conversation lengths by varying turn limits from 20 to 60 turns. Table 25 presents complete results across all evaluated turn limits using the test set (N=599). CogKC maintains consistent efficiency advantages across all turn limits, with performance metrics stabilizing by turn 25.

Performance metrics stabilize by turn 25: efficiency win rates plateau (50.4% at turn 25 vs. 52.7% at turn 50, only 2.3 percentage point increase for 25 additional turns), and gap resolution ratios show modest improvement (68.3% $\rightarrow$ 74.4%, only 6.1 percentage point gain for 25 additional turns). This diminishing returns pattern justifies our selection of turn 25 for primary evaluation, achieving 77.6% completion while maintaining strong

Table 24: Comprehensive Tutoring Effectiveness Statistics (RQ3, N=1,399). Full statistical validation with paired t-tests for continuous metrics (Efficiency, Gap Resolution Ratio). Win rates and completion rate reported descriptively.

	Absolute Performance Metrics			**Win Rates**		
Statistic	**Efficiency (gaps/turn)**	**Gap Res. Ratio (%)**	**Completion Rate (%)**	**Eff. Win Rate (%)**	**Gap Win Rate (%)**	**Tie Rate (Eff., %)**
CogKC	0.122±0.119	66.4±42.0	76.8	47.4	28.4	13.2
LLMKT	0.107±0.106	63.6±42.2	77.1	39.5	23.7	—
Difference	+0.015	+2.8	-0.4	+7.9pp	+4.7pp	—
95% CI	[0.007, 0.022]	[0.0, 5.6]	—	—	—	—
p-value	<0.001***	0.048*	—	—	—	—
Cohen's d	0.103	0.053	—	—	—	—
Test Statistic	t(1398)=3.837	t(1398)=1.982	—	—	—	—
Standard Error	0.0039	0.0141	—	—	—	—

$^{***}p < 0.001$, $^{*}p < 0.05$ (paired t-test)
Win rates and completion rate: descriptive statistics only (no hypothesis testing).
Gap Resolution Tie Rate: 47.9% (both methods resolve all gaps in 64.3% of conversations).

Table 25: Robustness Analysis Across Turn Limits (N=599 test set). All values shown as percentages. CogKC vs LLMKT comparison.

	Efficiency Win		**Gap Resolution**		**Completion**	
Turn	**CogKC**	**LLMKT**	**CogKC**	**LLMKT**	**CogKC**	**LLMKT**
20	43.2	34.4	55.9	52.0	61.8	60.1
25	**50.4**	**37.9**	**68.3**	**63.8**	**77.6**	**78.1**
30	52.3	38.9	72.9	68.4	86.6	88.0
40	52.2	37.3	73.5	68.4	95.2	95.3
50	52.7	37.6	74.4	69.3	98.0	98.0
60	52.3	38.3	74.7	69.8	99.0	99.0

Turn 25 (bold) selected for primary evaluation. Efficiency Win: % conversations where method wins on gaps/turn. Gap Resolution: Avg. % of identified gaps resolved. Completion: % conversations reaching resolution before turn limit.

discriminative power between methods. At longer turn limits (50-60), completion rates approach ceiling (98-99%) for both methods, reducing discriminative utility.

D.3.9 LLM-as-Judge Validation

We validated LLM gap resolution assessment against expert educator judgments. Human raters identified which knowledge gaps were completely resolved, enabling direct comparison with LLM judgments.

Inter-Rater Reliability. Inter-rater reliability on 21 shared conversations: Jaccard similarity = 0.536 (moderate), Cohen's κ = 0.189 (fair), Exact Match = 40.5%. Jaccard is preferred as it captures *which* specific gaps were resolved rather than just *whether* any gaps were resolved.

LLM-Human Agreement. LLM judgments compared against human raters show moderate agreement: Jaccard = 0.428 (shared conversations, n=21) and 0.452 (all conversations, n=60).

Gap resolution is the most challenging automated judgment task. This assessment depends on first identifying which gaps exist—a task where inter-human agreement is only moderate (Jaccard = 0.580 in RQ2). Gap resolution inherits this subjectivity while adding judgment about whether each gap was addressed, resulting in slightly lower inter-human agreement (Jaccard = 0.536). LLM-human agreement (0.428) reaches 80% of this baseline, comparable to the 84% ratio observed in RQ2's gap identification task. Given the compounded subjectivity, LLM judgments remain reliable for comparative evaluation.

E Chain of Evidence Analysis

To validate the causal claim that improved question targeting (RQ2) leads to improved tutoring effectiveness (RQ3), we test whether CogKC's advantage in RQ2 consistently translates to advantages in RQ3 outcomes across all 1,399 conversations. For each RQ2 criterion (LLM's holistic judgment considering both precision and coverage, precision, coverage) and RQ3 criterion (efficiency, gap resolution), we construct contingency tables and apply

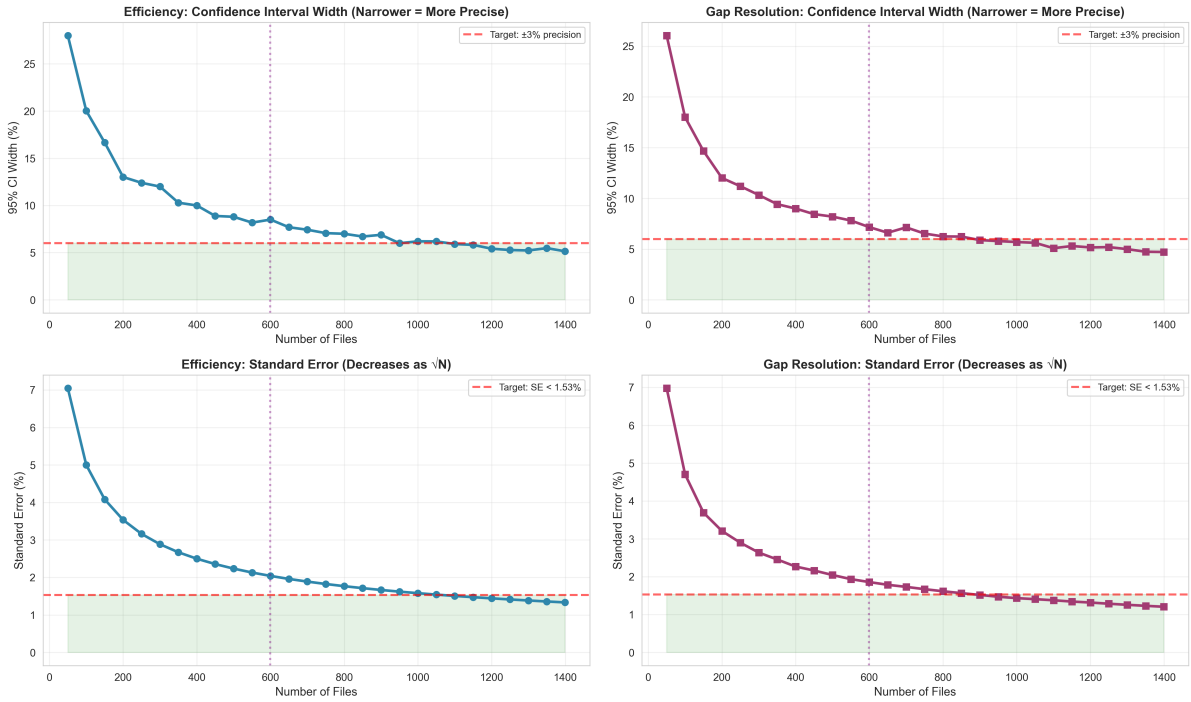

Figure 5: Statistical convergence analysis showing confidence interval widths and standard errors for efficiency and gap resolution metrics. Both metrics stabilize by N=1,100, with 95% CI widths below 6% (top row) and standard errors below 2 percentage points (bottom row). Vertical dashed line indicates stabilization point.

chi-square tests. We compute relative likelihood ratios: $\frac{P(\text{RQ3=CogKC}|\text{RQ2=CogKC})}{P(\text{RQ3=CogKC}|\text{RQ2=LLMKT})}$, quantifying how much more likely CogKC is to win on RQ3 given that it won on RQ2.

Table 26: Statistical validation results (N=1,399). All relationships show significant associations between RQ2 and RQ3 winners.

Chain	χ^2	**p-value**	**Ratio**
Holistic → Efficiency	16.69	0.0022**	1.20x
Holistic → Gap Resolution	25.49	<0.001***	1.30x
Precision → Efficiency	9.72	0.0454*	1.12x
Precision → Gap Resolution	14.40	0.0061**	1.24x
Coverage → Efficiency	16.84	0.0021**	1.26x
Coverage → Gap Resolution	36.60	<0.001***	1.66x

***$p < 0.001$, **$p < 0.01$, *$p < 0.05$.

All six relationships are statistically significant (all $p < 0.05$). The strongest relationship is Coverage → Gap Resolution (1.66x, $p < 0.001$): when CogKC wins on gap coverage in question generation, it is 66% more likely to win on gap resolution in tutoring outcomes. This reveals that comprehensive coverage of identified knowledge gaps in question design is the most critical factor for successful gap resolution during tutoring. Relative likelihood ratios range from 1.12x to 1.66x across all combinations, confirming that CogKC's advantages in question targeting (RQ2) consistently predict advantages in tutoring effectiveness (RQ3). Figure 6 provides detailed conditional probability analysis for all criterion combinations.

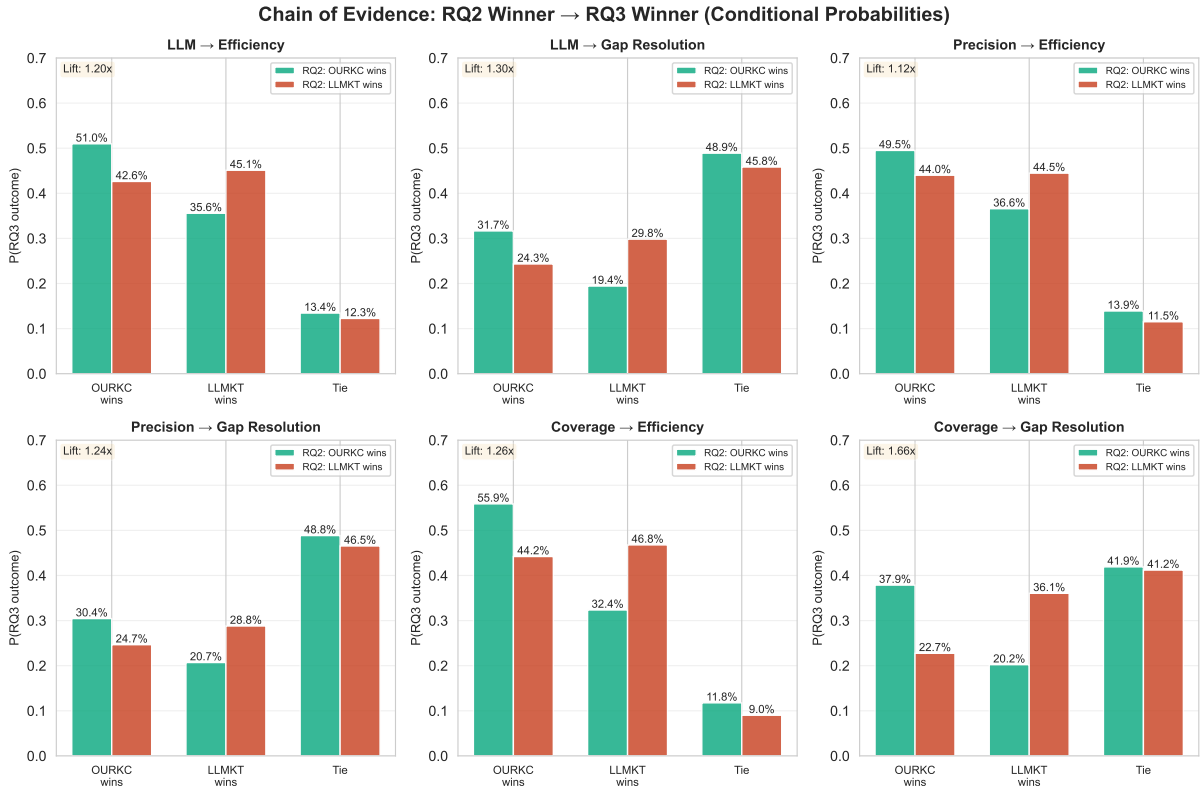

Figure 6: Conditional probabilities showing how RQ2 winners predict RQ3 outcomes across all criterion combinations. Green bars show probabilities when CogKC wins RQ2; red bars when LLMKT wins RQ2. Relative likelihood ratios indicate predictive strength.