Universiteit Leiden

Master Computer Science

Understanding Mathematical Misunderstandings: Enhancing LLMs' Support for Word Problem Solving

Name:	Kezhuoya Ma
Student ID:	2177854
Date:	28/11/2025
Specialisation:	Science Communication and Society
1st supervisor:	Dr.ir. D.J. Broekens
2nd supervisor:	Dr. M.J. van Duijn

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Math Word Problems (MWPs) require students to link textual information to quantitative reasoning. Learners often struggle with interpreting questions rather than with calculation. Current Large Language Model (LLM) benchmarks focus mainly on final answers, rarely examining the specific reasoning steps. This thesis aims to investigate how LLMs can support students' understanding of misunderstandings during math word problem solving. The research consists of six studies. First, we performed a baseline evaluation and an Actor–Critic investigation (Study 1) into state of the art MWP answer quality. Results show that the baseline model achieves high accuracy, although it occasionally produces inconsistent reasoning across repeated runs. Accuracy increased when the Actor–Critic approach reprocessed only incorrect answers. When applied to all items, however, it changed previously correct solutions. Second, we refined the GSM8K MWP dataset to remove ambiguous questions (Study 2). Third, we categorized potential misunderstandings underlying the errors made by children and LLMs respectively and identified the most frequent misunderstandings (Study 3). Fourth, we designed evaluation, classification, and guidance prompts based on these misunderstandings (Study 4). Finally, we integrated these components into the Math Coach prototype (Study 5) and conducted a pilot test (Study 6). The system successfully classified student inputs and routed them to specific feedback without revealing the final solution. These findings suggest that, under specific and controlled conditions, LLMs can move beyond answer generation to support the student reasoning process.

Keywords: Large language models; Math word problems; Misunderstanding categorization; Prompt design; Human–AI interaction

Contents

1 Introduction

1.1 Background and Motivation

Math Word Problems (MWPs) are central to elementary education because they require students to connect language comprehension with quantitative reasoning [1]. Previous research shows that students' errors usually stem from difficulties in interpreting information or failing to validate the result, rather than from arithmetic mistakes.

Researchers now extensively test Large Language Models (LLMs) to evaluate multi-step reasoning through MWP tasks. On standard benchmarks like GSM8K, LLMs have achieved human-level accuracy [2]. However, these high scores largely reflect how well LLMs' handle structured tasks shaped by specific techniques, such as chain-of-thought prompting. They reveal little about whether models genuinely understand problems in more realistic learning contexts [3, 4]. Because of this gap, accuracy metrics alone cannot ensure reasoning quality, as a correct answer do not guarantee a coherent reasoning process. Therefore, it is necessary to examine LLMs through a cognitive lens. Analyzing model errors and student misunderstandings clarifies the specific difficulties that the coaching tool must address. This analysis provides the basis for developing tools that support learning rather than merely reproducing correct answers.

1.2 Research Gap

Although LLMs have made steady progress in mathematical reasoning, most analyses still focus on results rather than on how reasoning unfolds, and thus lack a systematic account of the underlying process. Recent studies reveal that LLMs tend to skip intermediate steps or form unstable reasoning chains [5]. In contrast, educational psychology has long examined students' common error patterns in word problem solving, especially during comprehension and problem-solving phases 6. Yet these cognitive frameworks are rarely used to categorize LLMs' reasoning structures. As a result, we lack a clear classification of the specific student misunderstandings needed for targeted guidance. Meanwhile, most prompting research in educational contexts still aims to improve the quality of LLM outputs [4], but little work investigates how prompts shape students' reasoning or support their skill development. In classroom practice, LLMs mostly operate as solvers offering answers instead of engaging with students' comprehension or reasoning processes [7]. This limits their capacity to act as process-oriented coaches who help students review, adjust and explain their thinking. [8] argues that differences between model and human reasoning should be viewed not as limitations but as opportunities for joint exploration. However, this idea of productive divergences has seldom been applied to educational settings. Therefore, the design of human-model interaction that effectively supports process learning needs further investigation. This thesis examines a specific intersection: how can LLMs recognize and guide students' misunderstandings in mathematical word problems to support productive collaboration.

1.3 Research Questions

Based on these gaps, this study aims to investigate how LLMs can support students' understanding of misunderstandings during math word problem solving. Accordingly, it addresses the following research questions:

- **RQ1:** To what extent are LLMs capable of stable mathematical reasoning, and what is the impact of an Actor–Critic approach on reasoning quality?
- **RQ2:** What misunderstandings do LLMs and children respectively show when solving MWPs?
- **RQ3:** How can a prompt-driven interaction system be designed to support students solve MWPs in real tutoring settings?

By answering these questions, the study provide empirical findings regarding the effectiveness of prompt-based coaching and informs the design of learning-oriented support systems built on LLMs.

1.4 Research Contributions

This thesis followed a research path from model reasoning validation to educational application. We conducted six studies and made the following contributions:

- Study 1 (Baseline Evaluation and Actor-Critic Investigation): We evaluated the performance of state-of-the-art LLMs and tested the Actor-Critic approach to improve reasoning consistency.
- Study 2 (Benchmark Refinement): The GSM8K dataset was refined by removing ambiguous and mislabeled problems.
- Study 3 (Misunderstanding Categorization): We categorized the misunderstanding types of both children and LLMs across cognitive phases to inform prompt design.
- Study 4 (Prompt Design and Evaluation): Prompts were developed, including evaluation, classification, and guidance prompts.
- Study 5 (Prototype Development): These prompts were then integrated into a Math Coach prototype, which acts as a process-oriented scaffold.
- Study 6 (Pilot Testing): Finally, a pilot test was conducted to examine the feasibility of the system in educational settings.

Collectively, these six studies connect algorithmic analysis, cognitive understanding, and pedagogical application.

2 Related Work

This section reviews four key areas of research. This background is necessary to understand how Large Language Models (LLMs) can support students in Math Word Problems (MWPs) solving. Research on LLMs in MWP tasks outlines the capabilities and limitations of models' reasoning. Studies of children's misunderstandings describe cognitive challenges that appear during comprehension and solving. Prompting research indicates how prompts influence the stability and behavior of LLM outputs. Work on LLM-based educational agents discusses the roles models take in educational settings.

2.1 LLMs in Elementary Math Word Problems

Math word problems play an important role in elementary mathematics education. They are typically defined as textual descriptions of real-world scenarios requiring mathematical operations on numerical data [1]. They combine linguistic interpretation with quantitative reasoning. MWPs are therefore a key benchmark for assessing students' understanding. Recent progress in LLMs, including systems such as ChatGPT, has led to their application in MWP tasks. It is important to examine how well LLMs handle these tasks.

Existing evaluations show that LLMs perform well on many math word problems, especially those with clear linguistic cues and well-structured features. In mathematics education, such tasks are known as standard problems (s-problems), solved through arithmetic operations [9]. LLMs are less reliable on problematic problems (p-problems), which require drawing on realistic context [5]. State-of-the-art models (such as GPT-4, Gemini-1.5, and DeepSeek-V3) produce consistent reasoning and correct answers across multiple benchmarks [10, 11, 12]. Reasoning-oriented approaches (like chain-of-thought prompting [13], self consistency [14], and program-assisted reasoning [15]) further improve accuracy on structured datasets. For example, GPT-4 reaches nearly 90% accuracy on the GSM8k dataset, a benchmark of multi-step reasoning MWPs [2].

Despite these capabilities, analyses reveal several recurring weaknesses in LLMs' problem-solving processes. Li et al. [3] identify errors that range from basic calculation mistakes to gaps in reasoning, such as missing steps or inconsistent chains. Many of these failures stem from strategy selection errors rather than arithmetic difficulties. LLMs often rely on surface pattern matching or memorized templates [3]. They also skip the construction of a situation model or a mathematical model, moving directly to number manipulation without validity checks [5]. These tendencies resemble intuitive shortcuts in human reasoning but lack the flexibility needed for unfamiliar situations [16]. According to Strohmaier et al. [5] and Xie et al. [16], the dominance of standard problems in training datasets reinforces this bias. This emphasizes the final answer over the reasoning path. As a result, LLMs struggle to adapt strategies when facing unfamiliar situations.

Few studies link these errors to specific problem-solving phases. The CogMath framework [17] bridges this gap by assessing comprehension, solving and summarization. Its analysis shows that LLMs' mathematical abilities may be overestimated by 30-40%. Weaknesses appear across various problem-solving phases. The analysis suggests the importance of evaluations that reveal phase-specific difficulties rather than relying on an overall score. Opedal et al. [18] further report that LLMs reproduce several child-like cognitive bi-

ases. Ahn [19] argues that meaningful progress requires human-oriented interpretations that consider learners' needs, cognitive levels, and common misconceptions. Current LLM evaluations rarely adopt this perspective.

These findings show LLMs' critical reasoning flaws and the over-reliance on outcome-based metrics. These shortcomings confirm that raw LLM output functions primarily as a solver. It is thus unable to effectively aid in problem comprehension. This study focuses on designing a learning support system using a human-oriented framework to actively address specific shortcomings.

2.2 Children's Misunderstandings in Math Word Problems

For young learners, MWPs introduce both cognitive and linguistic demands, making them more than arithmetic exercises [20]. Research shows that successful solutions depend on accurate text comprehension, appropriate information processing, and reflection on the practical context [6]. Many misunderstandings arise from misinterpretation of the problem statement or focusing on irrelevant details rather than from calculation mistakes [21]. MWPs therefore serve as an effective diagnostic tool. They identify children's misunderstandings in both mathematical reasoning and contextual interpretation. This diagnostic potential has motivated extensive research work in mathematics education and educational psychology [6, 22, 23].

Since these errors reveal how children reason, researchers group them into phases and types using various frameworks [6]. One widely used framework is Newman's Error Analysis (NEA), which classifies observable mistakes into five phases: reading, comprehension, transformation, process skills, and encoding [24]. These phases reflect well-documented cognitive constraints in primary school learners. According to Piaget, most primary school students are in the concrete operational phases [25]. Their reasoning depends heavily on familiar contexts. Consequently, they struggle with abstract symbols and multi-step transformations. This makes the comprehension and transformation phases particularly prone to misunderstandings. Later studies further show that once a misunderstanding occurs early in the process, it often spreads to later steps. This results in unrealistic or incorrect final answers [6, 26].

Although these frameworks are informative, they differ widely in scope and level of detail. Most focus on observable performance and pay little attention to the cognitive causes behind children's mistakes. This limits their direct utility when designing pedagogical strategies for LLM–child collaboration. More recent work has attempted to simplify these phases to align them better with cognitive biases. For instance, Opedal et al. [18] describe three phases: text comprehension, solution planning, and solution execution. This structure provides a basis for analyzing the misunderstandings in LLMs and children.

These studies show that most classifications describe errors but lack insight into how misunderstandings unfold during actual problem solving. For my study, this gap specifies which aspects of children's reasoning require closer examination when designing the support structure.

2.3 Prompting in K–12 Mathematical Learning Contexts

In elementary mathematics classrooms, teachers often use questions to help students reorganize problem information and clarify quantitative relationships . These approaches strengthen students' comprehension [27]. Guiding questions prompt students to adjust their reasoning rather than rely on direct answers, which aligns with the metacognitive findings of [28]. The evidence suggests that encouraging students to revisit and verify their reasoning supports word-problem solving more effectively than providing additional calculation steps. Yet most studies focus on direct teacher-student interaction and offer little discussion on LLM implementation. A model's ability to offer this guiding support determines its capacity to engage meaningfully in student reasoning.

Prompt engineering research of K12 science, technology, engineering, and mathematics (STEM) contexts presents a different focus. The review of [4] shows that common prompting strategies include simple, zero-shot, few-shot, and chain-of-thought prompting. These strategies are often applied to problem solving, assessment and grading, task creation and feedback generation. Although they can improve model performance, they also show limitations: LLMs are highly sensitive to prompt wording, performance varies across tasks, and few studies validate these approaches with actual students. Beyond K–12 STEM research, [29] offers a broader categorization of prompting into technique-based prompting (e.g. structured or role-based prompting) and process-based prompting. The latter involve iterative refinement, feedback-driven design, and goal-oriented and task-specific frameworks. Process-based prompting is relevant for educational tasks that require precise evaluation and goal alignment. It emphasizes continuous calibration among task demands, model outputs, and human review. For math word problems, prompt design must adapt to student reasoning demands and maintain alignment with learning goals to remain reliable and applicable.

Most prompting research in educational contexts aims to enhance the quality of LLM outputs. Only a small number of studies examine how prompts influence students' reasoning or skill development. This gap is relevant for word-problem learning: prompts need to be stable on LLMs' side, but they must align with task-level learning goals. More importantly, prompts should be capable of addressing where students fail and guiding their reasoning challenges that underlie their mistakes.

2.4 LLMs as Collaborative Educational Agents

Researchers increasingly view LLMs as agents capable of memory, external tools, and planning to manage more complex tasks, moving beyond simple answer generation [30]. In education, LLM-based agents are grouped into two types. Instructional agents support teachers and students through classroom simulation, feedback and knowledge tracing, while domain-specific agents focus on subjects such as science, language, or professional training [30]. Math projects such as MathAgent [31] and MathChat [32] emphasize reasoning and error detection. Yet existing work still prioritizes accuracy over the educational roles these agents could take. This technical focus leaves a gap regarding how agents should engage pedagogically with learners.

LLMs often act as solvers, explainers, assistants and evaluators [7]. Even though their

roles have expanded, they still focus on producing answers over supporting reasoning. This risks passive learning, as students receive information without deep reflection. In contrast, educational research shows that a coach's questioning and feedback can trigger deeper reflection [33]. Grassucci [34] also notes that LLMs can function as patient tutors providing step-by-step guidance, or as collaborative partners encouraging strategy exploration. Building on this view, this study adopts the LLM as a coach, which combines tutor scaffolding with the partner strategy co-construction. In this framing, the coach role does more than deliver answers but aims to support students in developing habits of understanding, reasoning and reflection.

In artificial intelligence research, multi-module collaboration is often used to improve reliability. The Actor–Critic framework in reinforcement learning is one example, showing the complementary relation between generation and evaluation [35]. Although some educational have used in this idea, few focus on how such models should engage with learners. This dual-role structure inspires both algorithmic and educational implementation of this thesis. The Critic evaluates issues in responses, while the Actor generates feedback. This structure offers a conceptual reference for how humans and LLMs can collaborate effectively.

Beyond role-based and technical collaboration, a separate line of work considers how human reasoning interacts with model reasoning. Recent research suggests that differences in how humans and models conceptualize tasks create opportunities for joint exploration [8]. Productive divergence is described as a co-adaptive signal. However, existing work pays limited attention to how reasoning unfolds in education contexts. This thesis addresses this by examining their reasoning to design targeted guidance.

Current LLM-based agent designs remain model-centric. They offer limited evidence on how systems should interact when learner reasoning differs. This thesis addresses this interaction gap by developing a coach-like system for math word problem.

3 Methods

This chapter outlines the research methodology of the study, introducing the overall research method, research design, the framework for categorizing misunderstandings, and the experimental design. It also describes the dataset, implementation tools and ethical considerations.

3.1 Overall Research Approach

This study uses Math Word Problems (MWPs) to investigate how Large Language Models (LLMs) can recognize and support student misunderstandings. Since MWPs rely on text comprehension, they effectively reveal specific cognitive difficulties rather than simple calculation errors [21]. Likewise, LLMs frequently omit reasoning steps or produce hallucinations, despite performing well on structured tasks [3]. These errors provide the basis for analyzing their limitations in understanding.

To conceptualize this interaction, this study proposes a triangular model that positions LLMs, children and MWPs as three key elements (Figure 3.1). In this framework, $MWPs \rightarrow$

Children represent cognitive reasoning, $MWPs \rightarrow LLMs$ indicate algorithmic reasoning, and $Children \leftrightarrow LLMs$ illustrate the comparative and collaborative dimension.

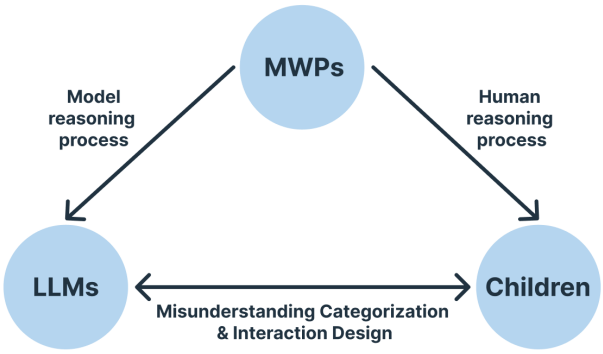

Figure 1: Triangular model of the relationships among LLMs, children, and MWPs

3.2 Research Design

The research process consists of six studies (as outlined in Section 1.4), structured to address the three research questions sequentially.

- **Addressing RQ1 (Algorithmic Reasoning):** Studies 1 and 2 focus on reasoning consistency and data quality. Study 1 evaluates baseline stability and tests the Actor-Critic approach for error correction. Study 2 refines the benchmark dataset by filtering ambiguous or mislabeled items. This ensures that subsequent evaluations focus on genuine reasoning errors of LLMs.
- **Addressing RQ2 (Misunderstanding Categorization):** Study 3 investigates specific reasoning failures in humans and models. We summarized common error patterns from literature [6, 3]. We then categorized these patterns according to the cognitive phases of comprehension, solving, and summarization. Based on this categorization, we inferred the specific misunderstandings underlying these observable errors.
- **Addressing RQ3 (Educational Application):** Studies 4, 5, and 6 translate the findings into a functional support system. Study 4 designs specific prompts for answer evaluation, error classification, and guidance. Study 5 integrates these prompts into the Math Coach prototype. Study 6 examines the system's feasibility through a pilot test with students.

3.3 Data and Implementation Basis

This study uses the GSM8K dataset [2], a benchmark for grade-school math word problems. We used the test set (1,319 problems) for model evaluation following the standard protocol. This dataset served as the primary source for model reasoning analysis and error identification.

We employed GPT-4o-mini model for baseline evaluation and Actor-Critic investigation. For prompt and tool design, GPT-5-nano was adopted to improve error evaluation and classification. We implemented all experiments through the OpenAI API to enable multi-turn dialogue. A lightweight prototype interface was developed using Gradio for user testing. The Python-based environment integrated data preprocessing, response evaluation and logic control.

3.4 Ethical Considerations

The pilot test was conducted with voluntary participants, obtaining informed consent from students and their guardians. All personal data was anonymized and used only for research purposes. The study design ensured that all tasks were age-appropriate and posed no risk or discomfort to participants.

4 Experiments

4.1 Study 1: Baseline Evaluation and Actor–Critic Investigation

Introduction

This study first aims to examine the reasoning stability and consistency of Large Language Models (LLMs) in solving Math Word Problems (MWPs) and to explore the potential of an Actor–Critic approach to improve such stability. Previous research shows that LLMs can perform well on structured tasks but still display systematic limitations. These biases often result from inappropriate strategy selection or hallucinated reasoning instead of arithmetic mistakes [3]. Similarly, children's misunderstandings often come from insufficient comprehension of problem information [1]. This suggests that effective problem solving relies heavily on semantic and structural understanding.

From an education perspective, a reliable tutoring model should maintain consistent and interpretable reasoning rather than simply providing correct answers. When LLMs produce contradictory reasoning chains, they cannot support trustworthy feedback and instructional guidance.

Therefore, the baseline performance of GPT-4o-mini was evaluated on the GSM8K dataset to establish a reference point. The Actor-Critic approach was then introduced to test whether the model could enhance reasoning consistency through self-evaluation and clarification. In this framework, the Actor generates clarified version of the problems, and the Critic reviews the reasoning and identifies potential issues. Together they form a cycle of evaluation, clarification, regeneration. By comparing the performance of the baseline and Actor-Critic approach across multiple runs, this study evaluates whether LLMs have the stability and reliability to act as a math coach.

Method

This section details the procedure and evaluation metrics for the baseline and Actor–Critic experiments.

The experiment used the GSM8K test set to avoid overlap with training data. We set the temperature to 0 for deterministic generation. For each run, we randomly sampled five few-shot examples to examine how stable the model's reasoning was under small contextual changes. Each output included reasoning steps and the final numeric answer.

We repeated the baseline experiment under the same settings to assess how consistent the model's reasoning was. While overall accuracy remained high, individual problem outcomes varied across runs. This suggested that the reasoning was not stable.

The Actor-Critic approach was then applied to test whether self-evaluation and clarification could make the reasoning more consistent. In this setup, the Critic reviews the initial response to identify potential issues, while the Actor rewrites the problem based on this feedback and solved it again. We tested two clarification modes:

- **AC on all answers:** The pipeline reprocessed every problem.
- **AC on wrong answers only:** The pipeline reprocessed the problems answered incorrectly in the baseline.

The experiment evaluated reasoning performance and consistency through three metrics:

1. **Total Accuracy:** The proportion of model answers matching the dataset answers, measuring overall task performance.
2. **Clarification Accuracy:** The accuracy achieved under the AC on all answers and AC on wrong answers only modes respectively, to compare how each strategy affected the results.
3. **Stable Correct Accuracy:** The proportion of problems solved correctly across all repeated runs, reflecting reasoning stability.

The procedure followed three steps. First, we ran the 5-shot baseline and recorded the output and overall accuracy. Second, we processed the dataset was then tested under the Actor–Critic approach in both modes. Finally, we calculated total accuracy, stable correct accuracy, and error flip rate to analyze the impact of AC.

Results

This study first ran the 5-shot baseline on the GSM8K test set to evaluate the reasoning performance of GPT-4o-mini. We selected the run achieving 93.1% accuracy as the reference for analysis. It generates a complete reasoning chain and allowed the extraction of both correct and incorrect subsets for further analysis. Based on this baseline, we tested two Actor-Critic clarification modes, AC on all answers and AC on wrong answers only, to compare how the scope of clarification influences reasoning outcomes. In addition, to examine reasoning stability, both the baseline and Actor-Critic approach were run twice independently under identical settings.

Table 4.1 shows the clear differences of the two clarification modes. The AC on wrong answers only mode increased accuracy by 3.11%. In contrast, the AC on all answers mode

decreased accuracy slightly by 0.23%. This indicates that reprocessing only incorrect answers improves outcomes. while clarifying all items may alter previously correct reasoning chains. Detailed results are listed in Table A.1 in Appendix A.

Mode	Accuracy	Accuracy Difference	Trend and Interpretation
AC on all answers	92.87%	Slight decrease (-0.23%)	46 *Correct→Incorrect* flips were observed, suggesting that unnecessary clarifications may disturb correct reasoning chains.
AC on wrong answers only	96.21%	Significant increase (+3.11%)	41 *Incorrect→Correct* conversions occurred without harming correct ones, indicating that AC on wrong answers effectively repairs reasoning errors.

Table 4.1: Consistency analysis of AC on all answers and AC on wrong answers only in this representative run

Table 4.2 presents the stability analysis. Even though overall accuracies were similar, the baseline achieved a higher stable correct accuracy (89.76%) compared to the Actor–Critic model (82.56%)

Approach	**5-shot Baseline**		**Actor–Critic**	
Iteration	**Run 1**	**Run 2**	**Run 1**	**Run 2**
Total Accuracy	91.89%	91.36%	86.73%	87.41%
Correct	1212	1205	1144	1153
Incorrect	107	114	175	166
Stable Correct Count	1184		1089	
Stable Correct Accuracy	89.76%		82.56%	
Stability	97.7%	98.3%	95.2%	94.5%

Table 4.2: Stability analysis of baseline and Actor–Critic approach

In summary, while the baseline model performs well, its reasoning consistency varies across runs. The Actor-Critic approach corrected reasoning errors when applied to incorrect items. When applied to all items, however, it disrupted previously correct solutions. Finally, the reliablity analysis shows that about 90 percent is solved correctly in two consecutive runs (Stable Correct Accuracy in Table 4.2), showing that the baseline method is relatively stable.

4.2 Study 2: Benchmark Refinement

Introduction

We manually inspected the incorrect answers to determine whether the errors resulted from the model's reasoning or issues with the questions themselves. During the manual inspection of incorrect subset, we observed that some responses marked as wrong resulted from dataset issues rather than model reasoning. Several items allowed more than one reasonable interpretation, yet the dataset accepted only one answer. As a result, a model could give a logically sound answer but still be judged incorrect. These cases affect evaluation results because they make it difficult to tell whether a mistake reflects a reasoning error or an ambiguity in the problem statement. To address this, we examined the incorrect subset of the baseline to identify and remove ambiguous and mislabeled items. This resulted in a refined version of the GSM8K dataset to serve as a clearer basis for later pedagogical experiments.

Method

The incorrect subset from the baseline run contained 91 problems. Each problem included the index, original problem, dataset answer, and model response. We manually reviewed these items one by one. A problem was labeled as ambiguity only when it met two conditions: (1) multiple interpretations were possible within the context; and (2) the ambiguity could not be resolved by the wording within the problem. If the dataset answer contradicted the problem statement, the item was tagged as mislabeled. All remaining items were considered genuine reasoning errors.

For example, in problem #404 of GSM8K, the problem asks: "Mel uses a 900-watt air conditioner for 8 hours a day. If he reduces the time he uses the air conditioner by 5 hours a day, how many kilowatts of electric energy will he save in 30 days?" The ambiguity comes from the phrase "reduces the time ... by 5 hours", which can be interpreted in two different ways:

Interpretation A ("reduces by 5 hours"): Mel only uses the air conditioner 3 hours a day, saving:

$$(8 - 3) \times 900 \text{ watts} \times 30 = 135 \text{ kWh}.$$

Interpretation B ("reduces to 5 hours"): Mel now uses the air conditioner 5 hours a day, saving:

$$(8 - 5) \times 900 \text{ watts} \times 30 = 81 \text{ kWh}.$$

Both interpretations are grammatically valid. Since the text provides no context to clarify the intended meaning, we labeled this item as ambiguous.

Problem #1188 shows another type of ambiguity. It asks: "How long will it take before he has 60 snowballs?" The problem does not specify the time unit, so both 5 hours and 300 minutes are valid descriptions of the same duration. Since the dataset accepts only one format, the model may be marked wrong even when its reasoning is correct. We therefore labeled such items as ambiguous.

Following this review, we extracted the indices for all ambiguous and mislabeled problems.

Then these problems are removed from the stable correct subset that includes all correctly solved problems in the two runs of baseline. Finally, the refined version of GSM8K was created for later experiments.

Results

Manual inspection identified 18 ambiguous problems and 2 mislabeled problems among the 91 items. The remaining 71 problems were kept as the genuine reasoning failures of the model.

After removing the 20 problematic items, the refined GSM8K dataset contains 1299 problems. The overall accuracy increased by 1.43% and about 1.5% of the original items were removed. Most ambiguous items involved quantitative relationships and a few were related to unit specification. The manual inspection inevitably involved some human judgment, but all decisions followed the predefined operational criteria.

This refinement shows that not all of all model errors come from weak reasoning. A small but significant portion of the errors was caused by unclear wording or incorrect labels in the dataset. Even though only a few items were removed, the change still had a clear effect on the overall accuracy. This suggests that ambiguous or mislabeled items introduce noise into evaluation. The model's actual reasoning ability can be reflected by removing them.

4.3 Study 3: Misunderstanding Categorization

Introduction

This study categorizes the specific reasoning failures of LLMs and children in math word problem solving based on the cognitive processes comprehension, problem solving and summarization. This analysis identifies the specific errors that the coaching system can focus.

When solving MWPs, both LLMs and children display a series of systematic reasoning failures [6, 3]. These failures are not limited to incorrect answers but they also reflect deeper difficulties in how information is interpreted, operations selected, and results validated. As discussed in Section 2, children's misunderstandings typically stem from intuitive reasoning and semantic misinterpretation, whereas LLMs' biases often arise from reliance on surface linguistic cues and a lack of realism checks. Although both may arrive at wrong solutions, their underlying processes differ. We developed the Misunderstanding Categorization to organize the misunderstandings of children and LLMs as a basis to inform the design of the Math Coach system.

Method

The categorization follows the cognitive phases of comprehension, solving, and summarization based on Liu et al. (2024). Additional reflection was added to the summarization phase to capture realism checks.

We constructed the categorization by integrating prior research on error analysis of children and LLMs. The process involved three steps: (1) identifying error types, (2) Inferring

the underlying misunderstandings, and (3) assigning them to cognitive phases. Children's misunderstandings were synthesized from [6]. LLM-related types were derived from error analyses reported in [3]. Both sets were organized within the three-phase structure (Appendix B).

Following [6], we categorized reasoning failures discussed in empirical research based on named mistakes, described difficulties, and recurring tendencies. Twenty-six representative subtypes were summarized and assigned to the cognitive phases. Table B.1 in Appendix B provides the complete classification, including definitions and representative examples. These subtypes were further consolidated into eleven mechanism-level categories. (Table B.2). The occurrences were qualitatively estimated based on reported patterns [6]. This helped assess the educational significance of each misunderstanding type.

For LLMs, nine error types of mathematical reasoning are defined by [3]. They used rule-based injections to simulate distinct reasoning error patterns, forming cognitive failure templates for repeated analysis. Building on these generation rules, the present study reformulated the nine reasoning-error patterns as corresponding misunderstanding types. Table B.3 in Appendix B summarizes the nine error types, definitions, inferred misunderstanding types, and examples from the paper.

Results

The categorization results show the distribution of misunderstandings between the cognitive phases. The types of misunderstanding between children and LLMs were organized within these phases. Table 4.3 visualizes the overall mapping.

Table B.2 lists the occurrences of children's misunderstandings derived from Verschaffel et al. [6]. This distribution shows which reasoning weaknesses are most educationally significant. We selected the misunderstanding types with the highest recurrence frequency among children as the focus for pedagogical application. Specifically, misinterpretation of quantitative relations and surface-driven operation selection formed the basis for the experimental design.

Overall, the Misunderstanding Categorization organizes specific reasoning failures across comprehension, solving, and summarization. These findings inform the next study which develops and tests prompts and prototypes.

4.4 Study 4: Prompt Design and Evaluation

Introduction

Study 3 identified two core misunderstandings: one linked to how children interpret information, and the other related to how they choose operations. In real interaction, since the system evaluates students' response not internal thoughts, these misunderstandings are examined through their observable patterns. This study therefore represented them as information-related error and operation-related error with a third category for responses that do not fit either pattern. The prompts were designed based on this categorization.

In the interaction, the model acts as a coach rather than a solver. It supports children's reasoning instead of providing answers or direct corrections. The goal is to prompt children

Cognitive Phase	Children's Misunderstanding Type	LLM Misunderstanding Type
Comprehension	• Relational mis-mapping • Representation construction failure • (other related comprehension issues)	• Misinterpretation of quantitative relations • Counting-range failure • Misrepresentation of unit relation or scale • Assumption beyond given information
Solving	• Rigid or misapplied schema activation • Intuitive operation bias • Failure to suppress intuitive response • Procedural reasoning discontinuity • Lack of strategic planning • Overgeneralization of proportional / rule-based reasoning • Weak metacognitive regulation / rule-driven transfer beyond context	• Inappropriate formula activation • Surface-driven operation selection • Incomplete reasoning sequence
Summarization	• Lack of realism check • Lack of metacognitive evaluation	• Lack of result validation • Inconsistency in reasoning integration

Table 4.3: Misunderstanding Categorization of LLMs and Children

to recheck their reasoning. This requests prompts with clear functions and constrained expression.

The system developed in this study requires the model to perform three specific tasks: evaluate understanding, identify the error type, and provide process-oriented guidance. Existing prompting taxonomies do not fully fit these functions. This study therefore designed three functional prompt types: evaluation, classification, and guidance. By comparing different phrasings and structures, the study selected more stable versions to support the system development introduced in the next section.

Method

The prompt design follows the idea of instructional scaffolding commonly used in elementary math teaching. Scaffolding relies on timely feedback to help students focus on the key steps. Based on this idea, this section describes the design process of the three prompts with the goal of producing stable and predictable outputs. To test the prompts, we created a set of simulated student responses covering common cases, including information-related errors, operation-related errors, incomplete reasoning, and irrelevant responses.

Evaluation prompts examine three aspects of students' response: (1) relevance to the problem, (2) presence of reasoning steps, and (3) evidence of basic understanding. We designed each prompt to make a single decision using a simple prompt structure. To obtain consistent yes/no outputs, the wording was adjusted and tested repeatedly across

different sample responses.

Classification prompts identify the primary error in a student's response so the system can route it to appropriate guidance. Three strategies were designed and tested:

- Prompt A (Open Label Generation): The model freely generates error labels in natural language without predefined constraints.
- Prompt B (Mutually Exclusive Categories): The model selects one label from three predefined options: INFO / OPERATION / OTHER.
- Prompt C (Separate Binary Judgments): The model answers independent yes/no questions about information use and operation choice.

Guidance prompts generate brief intervention to help students recheck their reasoning. Drawing on common classroom practices 6, the prompts were grouped into reflective, corrective, and guiding types. Each error label was linked to a corresponding guiding goal.

- *Information-related errors* $\longrightarrow$ *Reflective prompts*, which encourage students to revisit whether they used the required information correctly.
- *Operation-related errors* $\longrightarrow$ *Corrective prompts*, which prompt students to recheck their arithmetic steps and locate the possible mistake.
- *Other errors* $\longrightarrow$ *Guiding prompts*, which offer a general direction to help students clarify their thinking or refocus on the task.

All guidance prompts follow the same structure: a short statement ending with one question to continue the dialogue. They do not explain, correct or provide answers. This structure keeps the intervention light while still giving students a clear direction based on the detected error.

All prompt tests in Study 4 were run using GPT-5-nano. Preliminary comparison showed that GPT-5-nano produced more stable judgments and classifications than GPT-4o-mini, although its response speed was slightly slower.

Results

Multi-turn tests showed that prompts describing multiple tasks or containing layered constraints triggered inconsistent behavior. Some versions produced unnecessary explanations or generated full solutions directly. These cases indicate that prompts focusing on a single decision produced more stable results. Based on this, we separated the evaluation prompts into distinct tasks: relevance, presence of reasoning, and understanding. While this structure requires more API calls, it significantly improved output consistency compared to longer prompts.

Table C.1 in Appendix C summarizes the three classification strategies. Prompt A produced highly varied outputs, making it difficult to match predefined guidance paths. Even

for the same error, the model could generate different descriptions. For Prompt C, the two independent yes/no judgments sometimes contradicted each other, and some outputs required post-processing to decide the route. Prompt B was therefore selected as its mutually exclusive labels remained more consistent and could be routed directly.

Earlier guidance prompts versions sometimes revealed answers or solving steps. The final version avoids these issues: the model does not give solutions or judge the child's response. It built on the child's previous attempts using encouraging and non-judgmental tone, which fits the coach role.

This study designed and evaluated three types of prompts for math interaction: evaluation, classification and guidance prompts. The full prompt texts used are provided in Appendix D. Across tests, prompts that focused on one decision produced more consistent outputs. The classification strategy using mutually exclusive labels best supported clearer routing. The guidance prompts also aligned with the coach role maintaining light and supportive intervention.

4.5 Study 5: Development of the Math Coach Prototype

Introduction

This section introduces the integrated system, Math Coach, developed based on insights from the previous studies. The goal is to examine whether a controlled workflow can provide coach-like support when facing real problems and various student responses. The system is not a free-form conversational agent. Instead, it follows a structured workflow that manage how a child's response is evaluated, classified, and guided.

The workflow begins by presenting a random problem to the student. In the background, the system prepares the correct solution for later judgement. It then evaluates the response through sequential checks for relevance, reasoning, and understanding. Based on these results, the system classifies the error and routes it to targeted guidance. The full solution is revealed only after repeated unsuccessful attempts. These components allow the model to provide controlled, coach-like support rather than direct solutions or explanations.

Method

The figure in Appendix E presents the full system workflow. We used GPT-5-nano for the prototype. When a new interaction begins, the system randomly selects a problem from the refined dataset and displays it to the child. At the same time, the correct solution is pre-computed and cached in the background. It allows later evaluation to focus on the child's understanding of the target solution.

Once the child submits a response, the system evaluates if in the following order:

1. **Relevance Check:** The system first determines whether the reply connects to the math task. Off-topic messages trigger prompt that brings the child back to the problem.
2. **Reasoning Check:** If the response is relevant, the system checks whether any reasoning is shown. Pure numeric guesses lead to a prompt inviting the child to

describe their steps.

3. **Understanding Check:** For responses with reasoning, the system compares the child's interpretation with the pre-computed target to judge whether the child has understood the problem.

If the answer is relevant but shows misunderstanding, the system classifies it into information-related, operation-related, or other cases. Each label is mapped to a specific guidance. Information-related cases receive reflective prompts that encourage the child to revisit the problem details. Operation-related cases receive prompts that draw attention to arithmetic steps. Other cases receive general guiding prompts to help the child clarify their thinking.

After guidance is generated, the interaction history is updated and the attempt counter increases. When the child reaches three unsuccessful attempts, the interface reveals an optional button for asking the system to explain the solution. If clicked, the system gives a concise step-by-step explanation and moves on to the next problem. This prevents redundant loops while providing support when the child cannot progress.

To test whether the pipeline responded correctly to different response patterns, a set of child-like responses was created to intentionally trigger the main branches of the workflow, such as irrelevant replies, numeric guesses, and information- or operation-related misunderstandings.

Results

The workflow operated reliably across most test inputs. The evaluation module could distinguish irrelevant replies, identify answers lacking reasoning, and judged understanding of the target solution. The structure ensured that each condition was reviewed in order and that no step was taken too early or overlooked.

The classification module assigned the expected labels in the majority of cases and routed the input to the appropriate guidance. In a few borderline cases, responses were occasionally classified as OTHER label. These cases were directed to generic guiding prompts, which offered a safe fallback without disrupting the child's reasoning.

The guidance module maintained the coaching tone. Outputs avoided revealing answers, and built on previous attempts by drawing from recent interaction history. This prevented repetition and kept the dialogue focused on the current reasoning attempt.

Small inconsistencies appeared in both the evaluation and classification stages when the input was vague. They did not affect the overall interaction, as the structured workflow and fallback routes kept the system on track. This pattern reflects the sensitivity observed in Study 4 and shows why a fixed sequence of checks is necessary.

The attempt-based reveal method also behaved as intended. After three unsuccessful attempts, the explanation option appeared and generated a clear solution. The system then introduced the next problem.

Overall, the results indicate that the prototype can run through its full workflow and provide structured, coach-like guidance. While minor variation remains, the system handled

these cases safely. This confirms that the prototype is suitable for testing with children in the next study.

4.6 Study 6: Pilot Testing

Introduction

This pilot test implemented the prototype developed with a small group of Dutch-speaking children. The goal was to observe whether the system could run in real interaction and how children interacted with it. The focus was not on measuring learning effects, but on observing how children behaved when the system acted as a math coach, including their engagement, ease of expression, and any difficulties during interaction. In this test, children responded in Dutch. The pilot provides a basis for further system iterations and larger-scale user studies.

Method

Four students aged 10-12 participated in a classroom setting. Each student interacted with the prototype for around ten minutes and completed one or two math word problems.

To recruit participants for the pilot, information brochures in both English and Dutch were shared with schools. The brochures introduced the study purpose, participant criteria, and the expected procedure. A full copy of the English brochure is provided in Appendix F.

The problems were taken from the refined GSM8K subset and translated into Dutch. The guidance prompts and on-screen outputs were also presented in Dutch to match children's reading habits and reduced comprehension barriers. The evaluation and classification modules remained in English to avoid additional variation. Children typed their responses on a laptop via the Math Coach webpage. All interaction processes were recorded by the system, including interaction rounds, the children's inputs, evaluation and classification outcomes, and the routing results. These logs served as the main reference for analyzing system performance.

Sessions were conducted individually. Each child first read the problem on the screen and typed their response. After the system received the input, it ran the fixed pipeline and displayed the feedback. The child then continued the dialogue according to the guidance until they solve the problem or reached the attempt limit. The teacher was present but did not assist. The researcher monitored the technical setup and observed children's reactions. All inputs and outputs were logged for subsequent analysis.

Results

The following results were summarized from the system logs and on-site observations.

Children solved the problems within one to four turns. Most errors came from misinterpreting information in the problem, which aligns with the patterns identified in Study 3. They were generally able to understand the Dutch guidance and feedback. The guidance remained encouraging and did not reveal answers. Two of the four students occasionally looked toward the teacher but did not ask for help. All children showed focused attention and slight tension during the activity. Three students reported that the guidance was

helpful, while one found it was less useful. All of them showed visible impatience when the system responded slowly. Two children also needed extra time to type because they were unfamiliar with entering mathematical expressions on a laptop keyboard.

The pilot test indicated that the pipeline ran as intended. The modules connected smoothly and most Dutch inputs were processed and routed by the internal English evaluation and classification. The Dutch guidance was understandable and often led children to refine their reasoning. At the same time, several external factors influenced the experience. Response delays reduced engagement, keyboard input added operational effort, and teacher presence and classroom setting influenced how children interacted with the system. These observations provide concrete directions for the next round of refinement.

5 Discussion and Conclusion

5.1 Discussion

RQ1. To what extent are LLMs capable of stable mathematical reasoning, and what is the impact of an Actor–Critic approach on reasoning quality?

The results of Study 1 indicate that while LLMs often reach the correct final answer, their reasoning path varies across runs. As reported in [3], models do not consistently reproduce the same reasoning path for identical problems. However, overall in our case this did not results in a large drop in accuracy. The Actor-Critic experiments showed that AC is a double-edged sword. When AC was applied only to wrong answers, the model repaired a large amount of reasoning errors and overall accuracy increased. When AC was applied to all answers, it also altered correct solutions and reduced accuracy overall. These observations suggest that LLMs behave more reliably when focusing on fixing known mistakes, but become less reliable when re-checking correct reasoning without a clear target.

Study 2 results reveal that some error came from ambiguous or mislabeled items. Removing these items from the dataset ensured that the evaluation reflected actual model errors rather than problems with the dataset.

RQ2. What misunderstandings do LLMs and children respectively show when solving MWPs?

Study 3 present how LLMs' and children's misunderstandings are distributed across phases. These concentrated patterns help identify which steps in the problem-solving process that require the most attention. Because these steps often trigger misunderstandings, they offer practical focus for instructional support and guide the prompt and system design.

RQ3. How can a prompt-driven interaction system be designed to support students solve MWPs in real tutoring settings?

Study 4 and Study 5 examined how theoretical analysis was translated into practical support. The prompts were designed around specific pedagogical tasks and were tested and refined through repeated testing. This iterative approach aligns with the calibration

cycle proposed by [29] which stresses human-in-the-loop adjustments and task-aligned prompts improve reliability and educational fit in settings that require precise evaluation.

Testing results indicate that complex prompts often failed to evaluate responses correctly, added unnecessary explanations or revealed the solution. [4] notes that LLMs are highly sensitive to prompt wording and behave more consistently when the prompt handles a single task. We observed the same pattern, namely that slight changes in phrasing led to different outputs, particularly for the prompt that checked for correctness. The more detail we added to define correctness, the more difficult its behavior. In the end the prompt was simplified to ask the LLM to judge "whether the kid had understood the problem", and this was the best proxy for correctness.

Overall, the most consistent behavior occurred when each prompt required only one decision, so they were separated into independent actions. Some instability still remained due to the diversity of problems and responses. Recent systematic analyses such as [**?**] also shows that LLMs frequently struggle with these error types, so occasional variation in outputs is expected.

Study 6 shows that children understood the Dutch guidance and used it to continue the solution. This is relevant because [4] notes that prompt-based studies are rarely tested with real students. The main difficulties came from response delays and laptop keyboard input. These observations indicate that the system logic works but should be adapted to the interaction formats children normally use.

The key finding is that prompts need to remain interpretable, controllable, and tied to a specific misunderstanding for effective education support.

5.2 Limitations

The instability of LLMs' behaviors remains a source of concern. Reasoning chains differed across runs. The prompts were sensitive to small changes in wording. These patterns appeared in all studies, suggesting that the instability reflects a broader characteristic of current LLMs rather than one specific setting. In the end the prompts delivered, but significant prompt engineering was needed to get to this result.

There are also boundaries in the data and materials. The refined GSM8K subset may still contain ambiguous or mislabeled items, as manual cleaning involves subjective judgement.

The misunderstanding categorization relies on existing literature and does not cover different grade levels or language backgrounds.

The system is further constrained by its fixed workflow and language divergence. The pipeline cannot adapt to all response styles. The use of Dutch inputs with English evaluation modules may introduce additional noise.

The pilot test scale was small, involving only four participants. Their interaction was influenced by device handling and the classroom setting. As a result, the study demonstrates feasibility rather than instructional effects.

5.3 Future Work

Future research should explore the trigger boundary for the Actor-Critic approach, determining when to trigger AC and when to retain the original reasoning. The approach of AC should also be tested on other mathematics tasks beyond word problems.

The principles of dataset refinement can be made more systematic by developing more objective procedures for identifying ambiguous or mislabeled items.

The misunderstanding categorization can be broadened by using diverse student samples and different LLM types. This would allow it to adjust dynamically for varied applications.

Prompts and system design could focus on formal and identifiable error types, such as calculation mistakes or unit-conversion errors. Targeting these rigid errors may produce more consistent outputs than semantic misunderstandings. However, care should be taken not to overspecify prompts, because this may have adverse effects (as mentioned in the introduction).

The interaction format can be improved by reducing the barriers created by devices and input methods. Future studies on instructional effectiveness should be larger in scale and longer in duration. Experiments should avoid teacher presence and classroom pressure to better examine how students interact with the system in everyday learning situations.

5.4 Conclusion

This study addressed three research topics. It examined the stability of LLMs' reasoning on grade-level math word problems, categorized student and model misunderstandings, and finally iteratively designed the prompts and system prototype and conducted a small-scale pilot to evaluate feasibility.

The study produced several key findings. The model's reasoning consistency varied across repeated runs. It occasionally generated different reasoning chains for identical problems, suggesting that average accuracy alone is insufficient to verify reasoning quality, reliability is more important (and was reasonably good in our study).

In the Actor-Critic experiments, AC on wrong answers only helped repair errors and increased accuracy, while AC on all answers caused the model to change originally correct answers into incorrect ones, countering the positive effect.

Manual inspection of the model errors showed that that some errors came from ambiguous or mislabeled items rather than from model reasoning. We refined the GSM8K test dataset.

The categorization of misunderstanding highlighted two demanding steps: interpreting information and selecting operations, which guided the design of the prompts and pipeline . The pilot suggested that the system could complete the evaluation, classification, and guidance sequence, although response delay and the input format affected the interaction.

These results outline a workable link between misunderstanding categorization and prompt design. This structure allows LLMs to support children's problem solving without provid-

ing direct answers. The study also summarized that prompts behave more consistently when each prompt focuses on a single goal at a time. The prototype worked in real interactions and offers a starting point for further refinement and broader testing.

The scope of this study was still limited in several ways. It focused on one age group, one model version, and one type of task. The generalization of the results was constrained by prompt sensitivity and differences across model versions. The dataset refinement involved manual judgment, and the pilot had a small sample size. The findings should therefore be interpreted within these specific contexts.

In conclusion, this study examined how an LLM can act as a coach in children's problem solving under controlled conditions. Its effectiveness depended on clear trigger rules, stable prompt behavior, and an interaction format that children could manage. These observations provide concrete direction for future work on human–AI collaboration in mathematics learning.

References

[1] L. Verschaffel, B. Greer, and E. De Corte, *Making sense of word problems*. Lisse: Swets & Zeitlinger, 2000.

[2] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman, "Training verifiers to solve math word problems," 2021. [Online]. Available: https://arxiv.org/abs/2110.14168

[3] X. Li, J. Bai, Z. Zhang, J. Gu, and Y. Sun, "Evaluating mathematical reasoning of large language models: A focus on error identification and correction," 2024. [Online]. Available: https://arxiv.org/abs/2402.18668

[4] E. Chen, D. Wang, L. Xu, C. Cao, X. Fang, and J. Lin, "A systematic review on prompt engineering in large language models for k–12 stem education," *arXiv preprint arXiv:2410.11123*, 2024.

[5] A. R. Strohmaier, W. Van Dooren, K. Seßler, B. Greer, and L. Verschaffel, "Large language models don't make sense of word problems: A scoping review from a mathematics education perspective," 2025.

[6] L. Verschaffel, S. Schukajlow, J. Star, and W. Van Dooren, "Word problems in mathematics education: A survey," *ZDM–Mathematics Education*, vol. 52, no. 1, pp. 1–16, 2020. [Online]. Available: https://doi.org/10.1007/s11858-020-01130-4

[7] S. García-Méndez, J. C. Lázaro, P. Larrañaga, and C. Bielza, "A review on the use of large language models as virtual tutors," *Education and Information Technologies*, 2025, advance online publication. [Online]. Available: https://doi.org/10.1007/s10639-025-13137-6

[8] S. A. Gebreegziabher, "Cognition-inspired interactive frameworks for human–ai alignment," in *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM, 2025.

[9] L. Verschaffel, E. De Corte, and S. Lasure, "Realistic considerations in mathematical modeling of school arithmetic word problems," *Learning and Instruction*, vol. 4, no. 4, pp. 273–294, 1994.

[10] OpenAI, "Gpt-4 technical report," 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[11] Team Gemini, Google DeepMind, "Gemini 1.5: Unlocking multimodal understanding across vast knowledge domains," 2024. [Online]. Available: https://arxiv.org/abs/2403.05530

[12] DeepSeek AI, "Deepseek-v3 technical report," 2024. [Online]. Available: https://arxiv.org/abs/2412.19437

[13] J. Wei, X. Wang, D. Schuurmans, M. Bosma, E. H. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," 2022. [Online]. Available: https://arxiv.org/abs/2201.11903

[14] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," 2022. [Online]. Available: https://arxiv.org/abs/2203.11171

[15] L. Gao, A. Madaan, D. Zhou, D. Schuurmans *et al.*, "Pal: Program-aided language models," 2023. [Online]. Available: https://arxiv.org/abs/2211.10435

[16] W. Xie, S. Ma, Z. Wang, E. Wang, K. Chen, X. Sun, and B. Wang, "Do large language models truly grasp mathematics? an empirical exploration from cognitive psychology," 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2410.14979

[17] J. Liu, R. Xu, H. Guo, R. Xie, Z. Zeng, Y. Zhang, and J. Li, "Cogmath: Evaluating llms' authentic mathematical ability from a cognitive perspective," 2024, under review at International Conference on Learning Representations (ICLR) 2025. [Online]. Available: https://openreview.net/forum?id=x1nlO1d1iG

[18] A. Opedal, F. Rønning, A. Reifland, J. Tenfjord, and A. Lervåg, "Do language models exhibit the same cognitive biases in problem solving as human learners?" in *Proceedings of the 46th Annual Conference of the Cognitive Science Society (CogSci 2024)*. Rotterdam, The Netherlands: Cognitive Science Society, 2024.

[19] J. Ahn, R. Verma, R. Lou, D. Liu, R. Zhang, and W. Yin, "Large language models for mathematical reasoning: Progresses and challenges," *arXiv preprint arXiv:2402.00157*, 2024. [Online]. Available: https://doi.org/10.48550/arXiv.2402.00157

[20] G. Daroczy, M. Wolska, W. D. Meurers, and H.-C. Nuerk, "Word problems: A review of linguistic and numerical factors contributing to their difficulty," *Frontiers in Psychology*, vol. 6, p. 348, 2015. [Online]. Available: https://doi.org/10.3389/fpsyg.2015.00348

[21] J. B. Jaffe and D. J. Bolger, "Cognitive processes, linguistic factors, and arithmetic word problem success: A review of behavioral studies," *Educational Psychology Review*, vol. 35, p. 105, 2023. [Online]. Available: https://doi.org/10.1007/s10648-023-09821-6

[22] L. Menendez Cuervo, P. Nogueira, K. Baten, W. Van Dooren, and L. Verschaffel, "Task characteristics associated with mathematical word problem-solving performance among elementary school-aged children: A systematic review and meta-analysis," *Educational Psychology Review*, 2024. [Online]. Available: https://doi.org/10.1007/s10648-024-09954-2

[23] L. Verschaffel, "Taking the modeling perspective seriously at the elementary school level: Promises and pitfalls," in *Proceedings of the 26th Annual Conference of the International Group for the Psychology of Mathematics Education*, vol. 1, 2002, pp. 64–80.

[24] M. A. Newman, "An analysis of sixth-grade pupils' errors on written mathematical tasks," in *Research in Mathematics Education in Australia, 1977*, M. A. Clements and J. Foyster, Eds. Melbourne: Australian Council for Educational Research, 1977, vol. 1, pp. 239–258.

[25] J. Piaget, *The origins of intelligence in children*. W. W. Norton & Company, 1952. [Online]. Available: https://doi.org/10.1037/11494-000

[26] S. Pomalato, L. Ili, B. Ningsi, F. Fadhilaturrahmi, A. Hasibuan, and K. Primayana, "Student error analysis in solving mathematical problems," *Universal Journal of Educational Research*, 2020. [Online]. Available: https://doi.org/10.13189/ujer.2020.081118

[27] J. D. Stanton, A. J. Sebesta, and J. Dunlosky, "Fostering metacognition to support student learning and performance," *CBE—Life Sciences Education*, vol. 20, no. 2, p. fe3, 2021. [Online]. Available: https://doi.org/10.1187/cbe.20-12-0289

[28] I. Zeitlhofer, S. Hörmann, B. Mann, K. Hallinger, and J. Zumbach, "Effects of cognitive and metacognitive prompts on learning performance in digital learning environments," *Knowledge*, vol. 3, no. 2, pp. 277–292, 2023. [Online]. Available: https://doi.org/10.3390/knowledge3020019

[29] Y. Qian, "Prompt engineering in education: A systematic review of approaches and educational applications," *Journal of Educational Computing Research*, vol. 63, no. 7-8, pp. 1782–1818, 2025.

[30] Z. Chu, Y. Liu, H. Liu, J. Zhou, W. Y. Wang, and Z. Yu, "Llm agents for education: Advances and applications," *arXiv preprint arXiv:2501.04925*, 2025. [Online]. Available: https://arxiv.org/abs/2501.04925

[31] Y. Yan, S. Wang, J. Huo, X. Hu, and Q. Wen, "Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection," *arXiv preprint arXiv:2501.04696*, 2025. [Online]. Available: https://arxiv.org/abs/2501.04696

[32] Y. Wu, F. Jia, S. Zhang, H. Li, E. Zhu, Y. Wang, Y. T. Lee, R. Peng, Q. Wu, and C. Wang, "Mathchat: Converse to tackle challenging math problems with llm agents," *arXiv preprint arXiv:2306.01337*, 2023. [Online]. Available: https://arxiv.org/abs/2306.01337

[33] B. Hoffman and A. Spatariu, "The influence of self-efficacy and metacognitive prompting on math problem-solving efficiency," *Contemporary Educational Psychology*, vol. 33, pp. 875–893, 2008. [Online]. Available: https://doi.org/10.1016/j.cedpsych.2007.07.002

[34] M. Grassucci, A. Del Zozzo, V. Di Lollo, G. Orlando, A. Paladini, L. Penta, and A. Tarantino, "Beyond answers: How llms can pursue strategic thinking in education," *arXiv*, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.06713

[35] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3-4, pp. 229–256, 1992. [Online]. Available: https://doi.org/10.1007/BF00992696

Appendices

4

A Actor–Critic Experiments

Metric	AC on all answers	AC on wrong answers only
Initial Accuracy	93.10%	93.10%
Clarified Accuracy	92.87%	96.21%
Total Initial Correct	1228	1228
Total Initial Incorrect	91	91
Total Clarified Correct	1225	1269
Total Clarified Incorrect	94	50
Incorrect → Correct	43	41
Correct → Incorrect	46	0

Table A.1: Actor-critic performance comparison between two modes

B Framework Mapping Tables

Cognitive Phase	Error Type	Definition	Example
Comprehension	**1. Semantic structure misunderstanding**	Fails to distinguish semantic classes of addition and subtraction (change / combine / compare), causing operation confusion.	Treats "Tom has 3 more than Lily" as a change problem and computes 3 + 5 instead of 5 – 3.
	2. Misconstruction of situation model	Does not form a coherent mental representation linking quantities and relations.	Copies numbers from text and adds them without understanding their roles.
	3. Linguistic misunderstanding	Misreads complex sentence structure or comparative terms such as "less than," "after," "left."	Reads "after giving away 3 apples" as "after getting 3 apples."
	4. Over-focus on numerical data	Processes numbers in isolation without building semantic links.	Adds all numbers in the text regardless of context.
	5. Incomplete / inaccurate external representation	Drawings or models omit critical relations or contain errors that distort understanding.	Produces bar diagram with wrong length ratios between quantities.
	6. Teacher / textbook-induced simplified comprehension	Habitual expectation that every problem fits a standard template due to repetitive classroom formats.	Immediately searches for numbers to plug into a known formula without reading the story.
	7. Working-memory limitation	Limited capacity prevents integration of linguistic and numerical information into a coherent model.	Forgets earlier quantities while reading long sentences, causing inconsistent operations.
Solving	**8. Schema mis-selection**	Chooses an inappropriate problem schema during solution planning (premature schema activation).	Sees "more" → applies addition schema before verifying relationship type.

continued on next page

Cognitive Phase	Error Type	Definition	Example
	9. Wrong operation due to intuition	Chooses operation from linguistic cue or everyday intuition rather than logical relation.	In "Pete has 8 apples, gives away 3," child adds 8 + 3 instead of 8 – 3.
	10. Multiplication-always-makes-bigger fallacy	Believes multiplication always increases and division always decreases a number.	Multiplies 55 × 0.75 to get 41.25 but interprets it as "bigger because it's multiplication."
	11. Misapplied proportional reasoning	Extends proportionality to situations where it does not hold ("proportional bias").	"3 towels dry in 12 h; 6 towels → 24 h."
	12. Algorithmic fixation	Treats word problems as routine algorithmic tasks without conceptual reasoning.	Converts modeling problem directly into formula and computes mechanically.
	13. Lack of strategies planning	Begins calculation immediately without considering strategies or subgoals.	Starts computing before identifying what the question asks.
	14. Weak metacognitive regulation	Fails to monitor solution process or check path consistency (execution-stage self-monitoring failure).	Continues multi-step procedure even when intermediate results are illogical.
	15. Inhibition failure	Cannot suppress misleading intuitive rules (e.g., "more → add").	Applies addition for "than" even after teacher reminder it can mean subtraction.
	16. Implicit reasoning omission	Reasoning chain becomes fragmented because one or more inferential steps are skipped.	Skips intermediate reasoning and jumps to final computation without linking logic.
	17. Working-memory overload	High processing demand causes loss of information and computational slips.	Forgets previous step in multi-digit calculation and uses wrong number.
	18. Over-reliance on fixed schema / teacher models	Applies teacher-taught diagram or bar model mechanically without adapting to context.	Draws equal-group bars even when problem is a comparison type.

continued on next page

Cognitive Phase	Error Type	Definition	Example
Summarization	**19. Lack of realism check**	Does not test numerical answer against real-world knowledge or common sense.	Accepts "31 buses remainder 12" for 112 pupils (40 per bus).
	20. Non-realistic strategy / school-culture bias	Solves for the number expected by school culture rather than for a realistic outcome.	Ignores "impossible" answers because every school problem must have one exact number.
	21. Ignoring missing / inconsistent data	Performs calculations despite information gaps or contradictions.	Computes with incomplete data instead of questioning the problem.
	22. Lack of evaluation / reflection	Finishes without reviewing solution steps or reasonableness.	Writes final result immediately after first operation.
	23. Context neglect due to didactical contract	Classroom norms discourage questioning the problem statement (Brousseau 1997).	Never asks if numbers make sense because teacher expects a numerical answer.
	24. Teacher reinforcement of unrealistic reasoning	Teacher feedback rewards procedural accuracy over realism.	Gets full credit for numerically correct but unrealistic answer.
	25. Lack of transfer to authentic modeling	Cannot apply classroom methods to real-world situations.	Fails to use proportional reasoning in shopping context outside school.
	26. Limited meta-representational competence	Weak ability to reflect on and revise own representations (diagrams, models).	Draws incorrect graph and does not notice mismatch with numerical data.

Table B.1: Classification of children's misunderstanding subtypes across cognitive phases, summarised from [6]

Cognitive Phase	Consolidated Misunderstanding	Included Error Subtype	Occurrence
Comprehension	Relational mis-mapping	• Semantic structure misunderstanding • Over-focus on numerical data • Teacher / textbook-induced simplified comprehension • Linguistic misunderstanding	High
	Representation construction failure	• Incomplete or inaccurate external representation • Misconstruction of situation model • Working-memory limitation	Medium
Solving	Rigid or misapplied schema activation	• Schema mis-selection • Algorithmic fixation • Over-reliance on fixed schema / teacher models	High
	Intuitive operation bias	• Wrong operation due to intuition • Multiplication-always-makes-bigger fallacy	High
	Failure to suppress intuitive response	• Inhibition failure	Medium
	Procedural reasoning discontinuity	• Implicit reasoning omission	High
	Lack of strategic planning	• Lack of strategies planning	Medium
	Overgeneralization of proportional / rule-based reasoning	• Misapplied proportional reasoning • Working-memory overload	Medium
	Weak metacognitive regulation / rule-driven transfer beyond context	• Weak metacognitive regulation	Medium
Summarization	Lack of realism check	• Lack of realism check • Non-realistic strategy / school-culture bias • Context neglect due to didactical contract • Teacher reinforcement of unrealistic reasoning	High
	Lack of metacognitive evaluation	• Ignoring missing or inconsistent data • Lack of evaluation or reflection • Lack of transfer to authentic modeling • Limited meta-representational competence	Medium

Table B.2: Consolidated child-side misunderstandings, included subtypes, and qualitative occurrences across cognitive phases, summarised from the child error patterns reported in [6].

Cognitive Phase	Error Type (Abbr.)	Definition	Consolidated Misunderstanding	Example
Comprehension	Counting Error (CO)	Mistake in counting sequence or boundary inclusion/exclusion.	Counting-range failure	Counts "Sunday, Tuesday, Thursday" as 2 days instead of 3, producing the wrong average.
	Context Value Error (CV)	A wrong numerical value is extracted or substituted from the text.	Misinterpretation of quantitative relations	Takes "10 mph" as 20 mph when computing total time.
	Hallucination (HA)	Adds information not given in the problem statement.	Assumption beyond given information	Inserts an invented "+ 20 minutes delay on Tuesday," inflating total time.
	Unit Conversion Error (UC)	Applies an incorrect relationship between measurement units.	Misrepresentation of unit relation or scale	Converts "1 hour = 50 minutes," turning 2 hours into 100 minutes.
Solving	Operator Error (OP)	Uses an incorrect arithmetic operator while operands stay the same.	Surface-driven operation selection	Replaces "÷ 3" with "+ 3" in the final averaging step.
	Formula Confusion Error (FC)	Applies the wrong mathematical formula for the situation.	Inappropriate formula activation	Uses perimeter instead of area to compute wall-painting surface.
	Missing Step (MS)	Omits a necessary reasoning step that links prior and later steps.	Incomplete reasoning sequence	Deletes the final averaging step and reports the subtotal 168 as the answer.
Summarization	Contradictory Step (CS)	A later step contradicts or reverses an earlier conclusion.	Inconsistency in reasoning integration	A subsequent line reuses a previous result incorrectly, creating internal inconsistency.
	Calculation Error (CA)	The correct formula is used, but the arithmetic result is wrong.	Lack of result validation	In the "Tony goes to the store" problem, 168 ÷ 3 = 55 instead of 56.

Table B.3: Consolidated LLM misunderstanding types with definitions and examples, based on the LLMs' error patterns reported in [3].

C Prompt Experiments

Strategy	Prompt A – Open Label Generation	Prompt B – Mutually Exclusive Categories	Prompt C – Separate Binary Checks
Prompt	Identify the main error shown in the child's answer. Provide a short reason and end with one general label that captures the overall category of error, not a detailed subtype.	What is the main error shown in the child's answer? **INFO**: if only an error in the operand of the formula in the first wrong step when referencing the number in the question. **OPERATION**: if only an error in the operator of the formula used in the first wrong step. **OTHER**: if the child's response is a clarification question, statement, or case that does not fit INFO or OPERATION. Give a brief reason for your judgment and end with `[INFO | OPERATION | OTHER]`.	**C_OP**: Does the child's response contain only an error in the operand referenced from the question in the wrong step? Give a brief reason, then answer `[YES | NO]`. **C_INFO**: Does the child's response contain an error in the operand referenced from the question in the wrong step or formula? Give a brief reason, then answer `[YES | NO]`.
Output form	Natural-language label	Single categorical label	Two binary signals
Category constraint	None	Mutually restrictive, ensuring clear category boundaries	Independent, less constrained
Routing compatibility	Too fine-grained to map directly onto the predefined guidance	Can be directly routed to the corresponding guidance	Requires post-processing to determine final route
Consistency	Medium–High	High	Medium

Table C.1: Comparison of prompt styles for error classification

D Prompts Used in the Study

Relevance Check Prompt

```
You have to evaluate a child answering a math problem.
This is the math problem: "{question}"
The child answers "{child_answer}"
Is the child's reply connected to math problems in any way (number,
   calculation, partial attempt, clarification, request, an admission of not
   knowing)?
Answer [yes|no] as the last word.
```

Reasoning Step Check Prompt

```
You have to evaluate a child answering a math problem.
This is the math problem: "{question}"
The child answers "{child_answer}"
Does the child's response provide only a final numeric result or guess, stated
    directly as an answer, without posing a clarification question or showing
    any expression?
Answer [yes|no] as the last word.
```

Understanding Check Prompt

```
You have to evaluate a child answering a math problem.
This is the math problem: "{question}"
The child answers "{child_answer}"
{calculation}
Does the child's reply demonstrate understanding of the target quantity asked
   in the problem (not just partial values or intermediate steps) and give
   the value matches your calculation?
Answer [yes|no] as the last word.
```

Error Classification Prompt

```
You have to evaluate a child answering a math problem.
This is the math problem:"{question}"
The child answers "{child_answer}"
{calculation}
What is the main cognitive misunderstanding shown in the child's answer?
INFO: if only an error in the operand of the formula in the first wrong step
   when referencing the number in the question.
OPERATION: if only an error in the operator of the formula used in the first
   wrong step.
OTHER: if the child's response is a clarification question, statement, or case
    that does not fit INFO or OPERATION.
Answer [INFO|OPERATION|OTHER] as the last word.
```

Guidance Prompt — Information-related Errors

```
You are a collaborative learning coach helping a child solve a math word
   problem.
This is the problem: "{problem_text}"
The child answers: "{current_answer}"
Previous context: "{history_context}"
Encourage the child to revisit the problem and consider whether they have used
    the necessary information and correctly interpreted each detail.
Write one paragraph at most 2 short sentences total and end with one guiding
   question.
Do not say whether the answer is correct / provide the solution /restate the
   whole problem / list the steps.
Builds on the child's latest attempt and previous progress, avoid repeating
   earlier correct prompts.
Keep the tone encouraging and non-judgmental.
```

Guidance Prompt — Operation-related Errors

```
You are a collaborative learning coach helping a child solve a math word
   problem.
This is the problem: "{problem_text}"
The child answers: "{current_answer}"
Previous context: "{history_context}"
Encourage the child to carefully recheck their arithmetic steps to spot where
   the mistake might have happened.
Write one paragraph at most 2 short sentences total and end with one guiding
   question.
Do not say whether the answer is correct / provide the solution /restate the
   whole problem / list the steps.
Builds on the child's latest attempt and previous progress, avoid repeating
   earlier correct prompts.
Keep the tone encouraging and non-judgmental.
```

Guidance Prompt — Other Errors

```
You are a collaborative learning coach helping a child solve a math word
   problem.
This is the problem: "{problem_text}"
The child answers: "{current_answer}"
Previous context: "{history_context}"
Write one paragraph at most 2 short sentences total and end with one guiding
   question.
Do not say whether the answer is correct / provide the solution /restate the
   whole problem / list the steps.
Builds on the child's latest attempt and previous progress, avoid repeating
   earlier correct prompts.
Keep the tone encouraging and non-judgmental.
```

E System Workflow Diagram

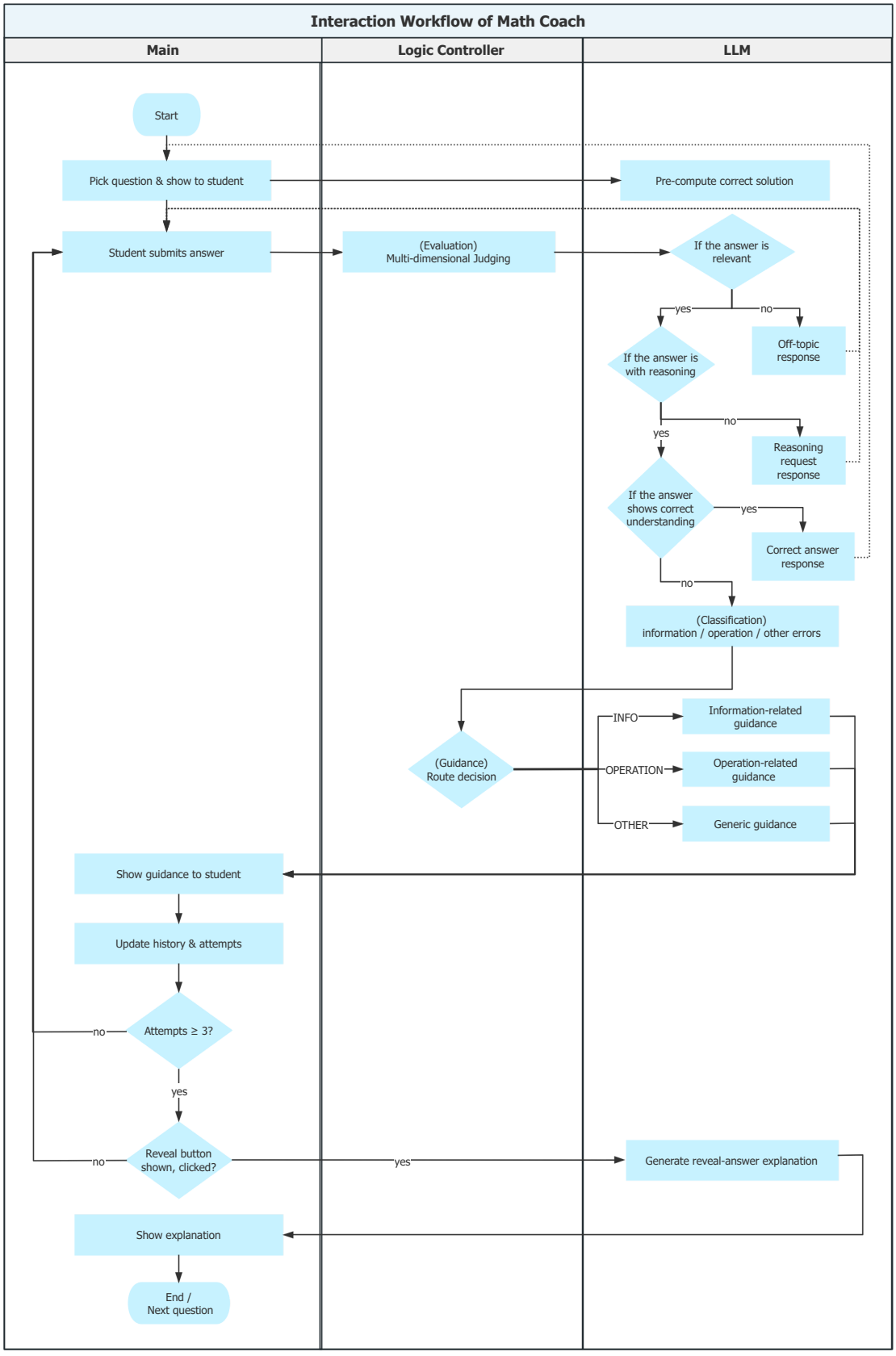

Figure 2: Interaction workflow of Math Coach

F Pilot Recruitment Materials

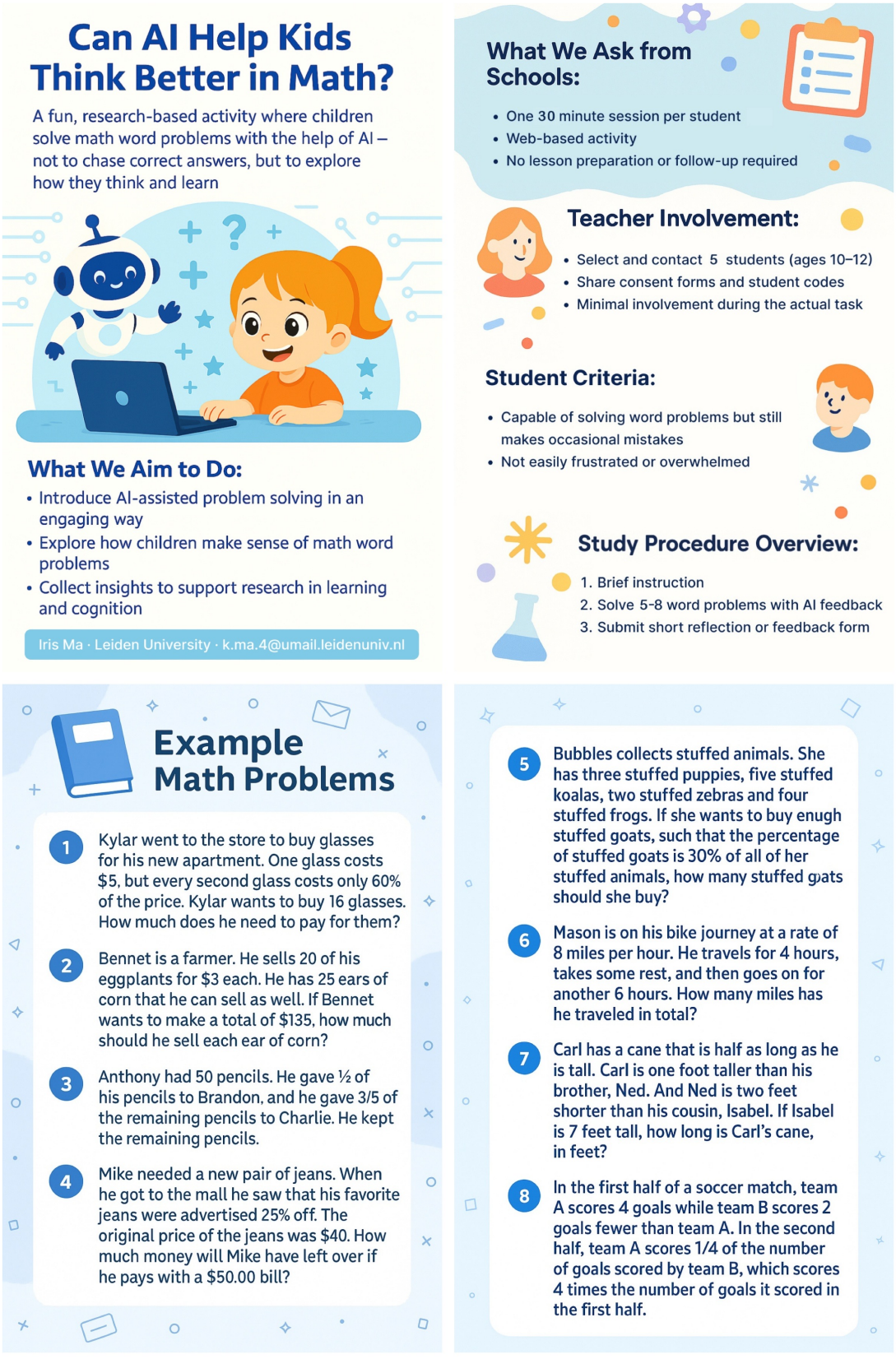

Figure 3: Pilot recruitment brochure.