THiNK: Can Large Language Models Think-aloud?

Anonymous ACL submission

Abstract

Assessing higher-order thinking skills in large language models (LLMs) remains a fundamental challenge, especially in tasks that go beyond surface-level accuracy. In this work, we propose THiNK (Testing Higher-order Notion of Knowledge), a multi-agent, feedback-driven evaluation framework grounded in Bloom's Taxonomy. THiNK frames reasoning assessment as an iterative task of problem generation, critique, and revision, encouraging LLMs to "think-aloud" through step-by-step reflection and refinement. This enables a systematic evaluation of both lower-order (e.g., remember, understand) and higher-order (e.g., evaluate, create) thinking skills. We apply THiNK to seven state-of-the-art LLMs and perform a detailed cognitive analysis of their outputs. Results reveal that while models reliably perform lower-order categories well, they struggle with applying knowledge in realistic contexts and exhibit limited abstraction. Structured feedback loops significantly improve reasoning performance, particularly in higher-order thinking. Qualitative evaluations further confirm that THiNK-guided outputs better align with domain logic and problem structure. The code of our framework, available at this anonymous link, provides a scalable methodology for probing and enhancing LLM reasoning, offering new directions for evaluation grounded in learning science.

1 Introduction

"Education is not the learning of facts, but the training of the mind to think."

— Albert Einstein

Assessing and enhancing large language models (LLMs) to support higher-order thinking (HOT) skills has become an emerging research focus (Latif et al., 2024; Xiao et al., 2025). As students increasingly rely on LLMs for flexible and accessible learning support, these models are being used as tutors to generate and solve complex mathematical problems that demand human-like HOT skills (Borge et al., 2024). All learners can progressively acquire HOT skills (Zohar and Dori, 2003), but the development requires continuous practice and guidance from knowledgeable and supportive educators (Saifer, 2018). One approach is the use of high-quality questions to cultivate HOT skills, thereby facilitating the assessment and promotion of students' cognitive development (Yao et al., 2021).

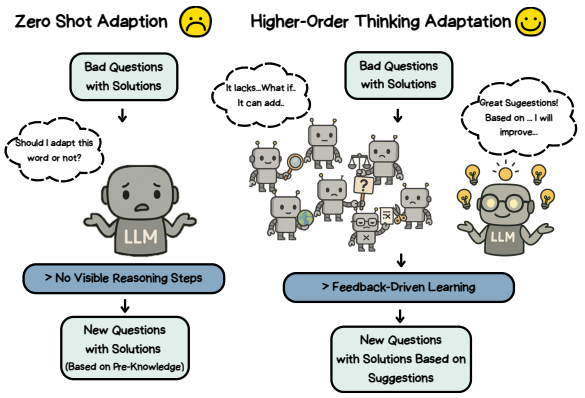

Figure 1: A figure shows the "think-aloud" process through iterative revision and reflection, a more robust assessment of HOT skills in LLMs.

Although recent studies have explored how to use LLMs to classify, generate, and solve math problems aligned with Bloom's taxonomy (He et al., 2023; Scaria et al., 2024a), none of them focus on the process of refining and regenerating problems, which limits our exploration and improvement of LLMs' HOT skills. Benchmarks like BIG-Bench Hard (Suzgun et al., 2022) aggregate performance across heterogeneous tasks, potentially masking critical differences in a model's proficiency at various cognitive levels (Srivastava et al., 2022). Moreover, LLMs often struggle with advanced reasoning processes (Collins et al., 2022), such as "thinking one step ahead" or adopting a theory of mind (ToM) perspective (Holterman and

van Deemter, 2023) to anticipate the types of problems that stimulate students' creativity or critical thought.

In this study, we focus on the cognitive task of refining and regenerating mathematical word problems (MWPs) as a process central to HOT skills (Widana, 2017), through refining initial attempts into more sophisticated outcomes iteratively (Fazey, 2010). Crafting high-quality MWPs to harmonize between abstract numerical reasoning, real-world context, and educational goals challenges the model to mimic and apply HOT skills (Testolin, 2024; Widana et al., 2018). As illustrated in Figure 1, we hypothesize that if an LLM can revise MWPs through iterative feedback integrations, it will reveal its underlying HOT skills through machine think-aloud. The think-aloud protocol is a widely used approach in cognitive psychology and learning sciences, in which participants articulate their thought processes in engaging real-time experimental tasks (Wolcott and Lobczowski, 2021). Now, researchers have used this approach in prompt frameworks (Chu et al., 2025).

To this end, we introduce THINK: Testing Higher-order Notion of Knowledge, a novel framework that aims to assess and improve the HOT capabilities of LLMs through the lens of mathematical problem generation. Unlike prior frameworks that rely primarily on accuracy-based metrics (Scaria et al., 2024b), THINK employs parallel evaluation agents grounded in Bloom's Taxonomy to assess models' capacity to iteratively review and revise flawed problems in response to structured feedback. This approach reflects real-world learning processes and provides a theoretically grounded, automated approach for probing the cognitive depth of LLMs, bridging natural language processing and educational theory. In summary, our contributions are threefold:

1. We present a multi-agent, feedback-driven evaluation framework grounded in educational theory. This novel framework empowers automated, structured evaluation of LLM reasoning and is further validated through qualitative evaluations by a human expert.

2. We introduce an iterative question refinement task to systematically probe a range of cognitive skills, from basic comprehension (e.g., *remembering* and *understanding*) to higher-order reasoning (e.g., *evaluating* and *creating*), including the generation of improved problems.

3. We conduct extensive experiments with multiple LLMs and establish a first-of-its-kind analysis of their reasoning performance across Bloom's levels, revealing key insights into their cognitive strengths and limitations.

2 Related Work

2.1 Cognitive Views on LLMs

Understanding the human-like cognitive capabilities of LLMs is essential in evaluating their potential for human-like linguistic abilities (Niu et al., 2024), including critical analysis and creative thinking. Existing studies have benchmarked LLMs across various cognitive dimensions, identifying similarities and divergences from human cognition. Srinivasan et al. (2023) pioneered the use of prototype analysis and understanding of proverbs to examine the commonsense reasoning of LLM. LLMs also demonstrate intuitive biases in psychological tests such as the Cognitive Reflection Test (Hagendorff et al., 2023) and show layer-specific alignment with neural signals in fMRI data (Zhang et al., 2024). Yet, key gaps remain, as LLMs often struggle with structured reasoning and inductive judgment, diverging from human-like patterns (Lamprinidis, 2023). Although techniques like chain-of-thought (CoT) prompting (Wei et al., 2022) can enhance model reasoning, they remain insufficient to capture higher-order cognition on a scale (Prystawski et al., 2022). Thus, current evaluation paradigms are heavily based on heuristics and lack standardized frameworks. This study addresses these limitations by examining whether LLMs can generalize beyond surface-level pattern matching to support deeper metacognitive competence.

2.2 Math Word Problem Generation

Existing MWP generation methods fall into four categories (i.e., template-based, rewriting-based, neural network-based, and LLM-based) (Kang et al., 2025). Template-based approach uses abstract skeletons, rewriting-based method modifies problem narrative descriptions and contexts, and neural network-based models the MWP generation end-to-end from topics and equation (Koncel-Kedziorski et al., 2016; Polozov et al., 2015; Zhou and Huang, 2019). These methods either fail to capture temporal efficiency and cognitive progression

during the generation process, or make it challenging to evaluate human-like reasoning (Amirizaniani et al., 2024). MWP generation can be used to explore more efficient proxy tasks as potential solutions. To address these shortcomings, we propose a feedback-driven, multi-agent framework based on LLMs to refine and regenerate flawed MWPs into high-quality ones with accurate answers. It naturally aligns with cognitive frameworks like Bloom's Taxonomy, demands structured reasoning, and has been employed in prior research (Scaria et al., 2024b) to investigate LLMs' abilities in generalization and metacognitive abilities.

3 THiNK

Our THiNK framework is grounded on educational foundations and designed to assess the extent to which current LLMs demonstrate HOT skills. In this section, we present the theoretical underpinnings that map constructs of human HOT skills onto LLMs' higher-order reasoning, alongside pipeline details.

3.1 Educational Foundations for Evaluation

Rather than evaluating LLMs solely based on surface-level correctness, our goal is to assess whether they can reason, generalize, and reflect in ways that align with human cognitive development (Ragab et al., 2024). Therefore, we draw on several key theories from the learning sciences and incorporate them into our THiNK framework.

Bloom's Taxonomy for LLMs The revised Bloom's Taxonomy (Krathwohl, 2002) categorizes cognitive processes into a hierarchical structure comprising lower-order thinking (LOT) skills (i.e., *remembering*, *understanding*, and *applying*) and HOT skills (i.e., *analyzing*, *evaluating*, and *creating*). The HOT skills of LLMs have been widely explored (Haase et al., 2025; Zhao et al., 2024) with research focusing on tackling complex tasks that challenge human performance. Although not exactly equivalent to human-like cognition, some scholars suggest that with sufficient interaction, LLMs could develop enhanced general intelligence and potentially advance toward a theory of mind or even rudimentary forms of consciousness (y Arcas, 2022).

Vygotsky's Zone of Proximal Development (ZPD) and Inquiry-based Learning According to Vygotsky and Cole (1978), the ZPD refers to the gap between the tasks that a learner can complete on their own and those they can successfully tackle when given targeted assistance from instructors (Shabani et al., 2010). Applied to LLMs, appropriate prompts are similar to the guidance of teachers, which can better instruct LLMs to think about disassembly and improvement, thereby triggering the HOT skills. Inquiry-based learning (Pedaste et al., 2015) emphasizes active engagement of learners in formulating questions and seeking answers. The ability to ask meaningful questions signals a transition from surface-level recall to deeper cognitive engagement (Yim and Su, 2025). Under this framework, the question generation serves as a measure of HOT skills and a mechanism to promote metacognitive reflection. We examine whether LLMs can simulate such inquiry behaviors, using their generated math questions as a proxy for reasoning depth.

3.2 Framework Implementation

The overview of THiNK is shown in Figure 2, which includes the data preparation stage, multi-agent evaluation structure, quality assessment protocols, and iterative revision loops, aiming to support comprehensive analysis of LLMs' cognitive performance.

3.2.1 Data Preparation

The foundation of our evaluation framework is built upon a curated collection of low-quality mathematical problems. Let $\mathcal{D} = \{p_1, p_2, ..., p_m\}$ denote our dataset of m mathematical problems, where each problem p_i consists of a question q_i and its solution s_i, i.e., $p_i = (q_i, s_i)$. We construct $\mathcal{D}$ from two primary sources. The first subset, $\mathcal{D}_{\text{bad}}$, contains $m_1 = 20$ poorly constructed problems crawled from social media platforms (e.g., Reddit, Twitter). These examples exhibit deficiencies in pedagogical soundness and fail to satisfy core quality criteria. The second subset, $\mathcal{D}_{\text{syn_bad}}$, comprises 100 synthetically generated questions produced by GPT-4o using prompts detailed in Appendix B.1. These questions mimic the structural weaknesses of $\mathcal{D}_{\text{bad}}$ by deliberately omitting the "Five Keys" components, defined in Appendix A.1.

3.2.2 Multi-Agent Evaluation Structure

Algorithm 1 details the implementation of this framework, which employs a parallelized multi-agent system $\mathcal{A} = \{A_1, A_2, ..., A_7\}$, where each agent A_j for $j \in \{1, ..., 6\}$ corresponds to a spec-

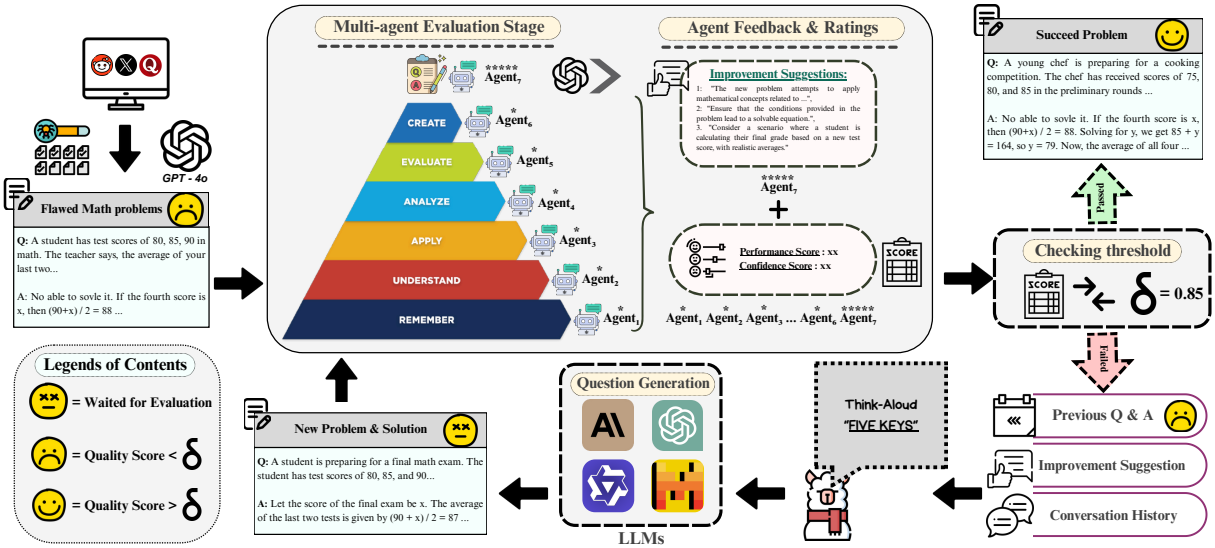

Figure 2: Overview of the THINK. The pipeline begins with flawed math problems 😣 that are iteratively refined. The core multi-agent evaluation stage uses six Bloom-aligned agents and one heuristic agent to assess quality, providing scores and targeted feedback. Guided by the "Five Keys" and prior suggestions, LLMs revise or generate new problems via a think-aloud process. A quality threshold determines success 😊 or triggers further refinement.

ified cognitive level in Bloom's Taxonomy, and A_7 represents a holistic language and pedagogical evaluation. Given a problem $p_i \in \mathcal{D}$, each agent A_j generates the tuple:

$$A_j(p_i) = (PS_j(p_i), CS_j(p_i))$$

where $PS_j(p_i) \in [0, 100]$ is the performance score and $CS_j(p_i) \in [0, 100]$ is the confidence score. The detailed prompts of each agent are provided in Appendix B.2. These are produced using CoT prompting, encouraging explicit, step-wise reasoning (Wei et al., 2022) aligned with the agent's cognitive level. In addition, the holistic evaluation agent A_7 further outputs an improvement suggestion:

$$A_7(p_i) = (PS_7(p_i), CS_7(p_i), IS(p_i))$$

where $IS(p_i)$ provides structured feedback on how to improve the problem. The prompt used by the holistic agent is provided in Appendix B.3. This feedback assesses whether the problem satisfies the "Five Keys" components, evaluates lexical and syntactic complexity, and identifies ambiguities or unsolvable elements. It examines the alignment between the problem and its proposed solution strategy (See Figure 2).

3.3 Quality Assessment Protocol

We define three core metrics to assess the quality evolution from a given problem p_i to its revised version, collectively capturing correctness, inter-agent consistency, and confidence:

Pass Rate (PR):

$$PR(p_i) = \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} \mathbf{1}(PS_j(p_i) > \tau)$$

where τ is the predefined passing threshold. Following Zheng et al. (2023), we adopt a reference-guided rating approach in which each agent assigns a performance score based on specific evaluation criteria. The pass rate reflects the proportion of agents who consider the problem sufficiently well-constructed to meet their level-specific standards.

Agent Agreement (AA):

$$AA(p_i) = \kappa \cdot (\{b_j(p_i) \mid j \in \{1, ..., |\mathcal{A}|\}\})$$

with $b_j(p_i) = \mathbf{1}(PS_j(p_i) > \tau)$ as a binary indicator. $\kappa(\cdot)$ denotes Cohen's Kappa coefficient, quantifying the agreement between agents beyond chance and reflecting evaluation consistency (Cohen, 1960).

Average Confidence (AC):

$$AC(p_i) = \frac{1}{|\mathcal{A}|} \sum_{j=1}^{|\mathcal{A}|} CS_j(p_i)$$

This aggregates how confident the agents are in their evaluations. It serves as an indicator of reliability, consistent with findings from recent work on trust calibration in LLM outputs (Jung et al., 2024).

Algorithm 1 THINK Framework

Require: Problem set $\mathcal{D} = \{p_1, \ldots, p_m\}$, agents $\mathcal{A} = \{A_1, \ldots, A_7\}$, threshold τ, weights (α, β, γ), maximum iterations R

Ensure: Improved problem set $\mathcal{D}_{\text{improved}}$, final cognitive performance scores $\mathcal{Q}_{\text{final}}$

$\mathcal{D}_{\text{improved}} \leftarrow \emptyset$, $\mathcal{Q}_{\text{final}} \leftarrow \emptyset$
for each $p_i \in \mathcal{D}$ **do**
  $r \leftarrow 0$, $success \leftarrow$ **False**
  while $r < R$ **and not** $success$ **do**
    Evaluate p_i using all agents $\mathcal{A}$ to obtain scores (PS, CS) and feedback IS
    Compute $PR(p_i)$, $AA(p_i)$, $AC(p_i)$, and composite quality score $Q(p_i)$
    if $Q(p_i) > \tau$ **then**
      $success \leftarrow$ **True**
    else
      ▷ Refine via feedback → Think-aloud
      $p_i \leftarrow \text{LLM}(p_i, IS)$
      $r \leftarrow r + 1$
    end if
  end while
  Add final version of p_i to $\mathcal{D}_{\text{improved}}$
  Add final $Q(p_i)$ to $\mathcal{Q}_{\text{final}}$
end for
return $\mathcal{D}_{\text{improved}}$, $\mathcal{Q}_{\text{final}}$

Success Criterion: We combine the three metrics into a composite quality score:

$$Q(p_i) = \alpha \cdot PR(p_i) + \beta \cdot AA(p_i) + \gamma \cdot AC(p_i)$$

In our setting, $\alpha = 0.5$, $\beta = 0.3$, and $\gamma = 0.2$ are weights determined by expert tuning. Finally, a problem is deemed successful if:

$$\text{Success}(p_i) = \mathbf{1}(Q(p_i) > 85)$$

We choose these three metrics to assess problem quality from different dimensions: PR measures correctness across cognitive dimensions, AA checks for consistent evaluation beyond random agreement, and AC incorporates evaluators' confidence in their judgments. Hence multi-agent structure provides a robust assessment, which is essential for evaluating higher-order reasoning.

3.4 LLM Think-aloud and Pipeline Overview

The structured pipeline enables LLMs to refine flawed math problems using agent-generated feedback iteratively. Grounded in the educational theories discussed in Section 3.1, the process incorporates a think-aloud protocol (Wolcott and Lobczowski, 2021), a widely used approach in cognitive psychology and learning sciences, in which participants articulate their thought processes in real time while engaging in experimental tasks, particularly those involving learning and problem solving. In this study, LLMs act as participants in self-reflective revisions and demonstrate their thinking processes based on agent feedback. When a problem p_i fails to meet the quality threshold, it undergoes iterative refinement. The holistic agent A_7 provides structured feedback $IS(p_i)$, which is returned to the LLM to generate an improved version of the problem. The revised problem is then re-evaluated by all agents. This loop continues for up to R iterations until the quality score exceeds the threshold. The algorithm details are provided in Algorithm 1. Each iteration promotes improvement in question quality, also allowing us to examine the LLM's reasoning and revision behaviors. This enables a deeper analysis of both lower- and higher-order thinking capabilities.

4 Experiment

We present experiments conducted with the THINK framework using the dataset introduced in Section 3.2.1, a curated collection of web-crawled and synthetic flawed math problems designed to assess LLMs' reasoning and revision capabilities.

4.1 Metrics

Cognitive Performance via Bloom's Evaluation The cognitive performance of LLM is evaluated using Bloom's taxonomy within our multi-agent evaluation framework, with each agent denoted as A_1 through A_6. Beyond raw performance, we analyze score improvements across iterations as a proxy for the model's revision ability and depth of reasoning. Given the potential unreliability of subjective performance scores, we incorporate an additional final quality check aligned with the objective scoring protocol described below. To further ensure the reliability of the framework, we conduct a qualitative comparison between zero-shot question revisions and those guided by THINK.

Quality Performance Evaluation We define two metrics to evaluate the effectiveness of the iterative refinement process within the THINK framework.

`RoundsToPass` Denoted as $R_{\text{pass}}(p_i)$, this metric measures the efficiency of the refinement loop by

Model	Remembering	Understanding	Applying	Analyzing	Evaluating	Creating	Avg.
GPT-4O	86.92 ↑ 26.92	**82.96** ↑ 5.79	**76.71** ↓ 0.46	**83.50** ↑ 4.21	**83.54** ↑ 2.92	**82.62** ↑ 4.21	**82.71** ↑ 3.51
GPT-4O-MINI	85.21 ↑ 15.12	**82.96** ↑ 0.71	74.50 ↓ 5.88	82.88 ↓ 1.38	83.08 ↓ 1.88	82.42 ↓ 0.54	81.51 ↑ 1.91
GPT-3.5-TURBO	82.29 ↑ 12.96	81.25 ↑ 1.29	71.83 ↓ 6.12	81.12 ↑ 0.54	80.92 ↓ 0.54	80.25 ↑ 0.92	79.61 ↑ 2.41
QWEN2.5-14B-IT	90.92 ↑ 42.50	74.92 ↑ 2.42	71.54 ↑ 0.71	81.25 ↑ 7.46	81.88 ↑ 6.17	77.83 ↑ 5.92	79.39 ↑ 2.49
QWEN2.5-7B-IT	**91.96** ↑ 34.25	72.54 ↓ 0.54	68.54 ↓ 5.79	76.96 ↓ 0.21	78.38 ↑ 0.33	73.88 ↓ 0.12	77.38 ↑ 0.18
MISTRAL-8B-IT	91.62 ↑ 35.79	67.96 ↓ 3.21	66.92 ↓ 6.21	74.75 ↑ 0.50	76.21 ↓ 0.33	70.33 ↓ 3.04	74.30 ↓ 2.90
LLAMA-3.1-8B-IT	90.42 ↑ 30.38	71.58 ↓ 3.75	69.04 ↓ 6.58	78.08 ↑ 1.83	77.58 ↓ 0.75	75.08 ↓ 0.21	76.80 ↓ 0.40
Average	88.48 ↑ 28.42	76.02 ↑ 0.96	71.15 ↓ 4.19	79.22 ↑ 1.71	80.80 ↑ 1.45	77.20 ↑ 1.30	78.81 ↑ 1.16

Table 1: Model performance across the six cognitive levels defined by Bloom's Taxonomy. Each cell reports the average score for the corresponding cognitive category, with ↑ and ↓ indicating the relative improvement or decline compared to the previous round, based on the model's revision. **Bold** and underline highlight the best and second-best performances.

recording the number of iterations required for a problem p_i to exceeds the quality threshold τ:

$$R_{\text{pass}}(p_i) = \min \left\{ r \in [1, R] \mid A(p_i^{(r)}) > \tau \right\},$$

where $A(p_i^{(r)})$ is the quality score at iteration r, R is the max number of allowed refinement rounds.

`AvgQualityScore` Denoted as Q_{avg}, this metric captures the average quality across all refinement steps:

$$Q_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{R_i} \sum_{r=1}^{R_i} A(p_i^{(r)}) \right),$$

Together, these metrics provide a holistic view of the model's iterative reasoning behavior, including its ability to improve question quality, engage with structured feedback, and maintain consistency in producing high-quality outputs.

4.2 Evaluated Models and Settings

We evaluate a set of off-the-shelf LLMs using THINK framework to probe their capacity for higher-order reasoning capabilities. We include four open-source models: LLAMA-3.1-8B-IT (Llama, 2024), Mistral-8B-IT (Jiang et al., 2023), QWEN2.5-7B-IT, and QWEN2.5-14B-IT (Qwen2.5, 2025); and three closed-source models: GPT-3.5-TURBO, GPT-4O-MINI, and GPT-4O (OpenAI, 2024). Notably, GPT-4O is used to implement the multi-agent roles within our pipeline, with the temperature set to 0, following its strong cognitive reasoning performance in the current benchmark (Huang et al., 2024). All open-source models are run on two NVIDIA A6000 GPUs (32GB), and the experiments involving OpenAI models incur a cost of approximately $300.

Model	$R_{\text{pass}}(p_i)$	Q_{avg} (%)
GPT-4O	2.35	82.46 ↑ 0.10
GPT-4O-MINI	2.57	78.68 ↑ 0.01
GPT-3.5-TURBO	2.60	73.46 ↑ 0.02
QWEN2.5-14B-IT	2.08	77.10 ↑ 0.11
QWEN2.5-7B-IT	2.12	72.47 ↑ 0.05
MISTRAL-8B-IT	2.04	72.05 ↑ 0.06
LLAMA-3.1-8B-IT	2.17	71.11 ↑ 0.03
Average	2.27	75.76 ↑ 0.05

Table 2: Performance of LLMs on iterative refinement tasks. $R_{\text{pass}}(p_i)$ is the average number of refinement rounds required for problem p_i to exceed the quality threshold. Q_{avg} represents the final quality score after all refinement steps. ↑ indicates the relative improvement in quality score compared to the last iteration.

4.3 Experimental Results

Table 1 and Table 2 show the performance of LLMs on the THINK framework, covering both cognitive skill levels defined by Bloom's Taxonomy and metrics for iterative refinement. We highlight several observations as follows:

LLMs Underperform in Mid-Level Cognitive Domains Table 1 shows that LLMs achieve consistently high scores in lower-order reasoning tasks such as *Remembering* and *Understanding*, indicating strong capabilities in information recall and paraphrasing. However, there is a marked performance drop in the *Applying* category, which requires transferring learned concepts to a real-world scenario. Nearly all models exhibit degradation in this dimension, suggesting that while LLMs are effective at surface-level understanding, they struggle to deploy knowledge in practical

Version	Question	Solution
Original	An orchestra of 120 players takes 40 minutes to play Beethoven's 9th Symphony. How long would it take for 60 players to play the symphony?	(Implied): Assumes inverse proportionality, suggesting it would take 80 minutes for 60 players.
Zero-shot	A school band with 120 members plays a song that lasts 40 minutes. If the same song is played by a band with 60 members, how long will the performance last?	It will still take 40 minutes for 60 band members to perform the song.
THINK-Guided	An orchestra of 120 musicians performs Beethoven's 9th Symphony in 40 minutes. Assuming equal contribution, how long would it take 60 musicians to complete the same symphony?	Since performance duration does not depend on the number of musicians (as long as all parts are covered), it would still take 40 minutes for 60 musicians.

Table 3: Comparison of the original flawed problem and its improved versions via zero-shot prompting and the THINK framework, demonstrating enhanced domain-appropriate reasoning. Sampled from QWEN2.5-14B-IT.

or problem-solving contexts. Even GPT-4O, the top-performing model, demonstrates a noticeable decline in this category, crafting a cognitive gap between comprehension and execution.

LLMs Are Not Always Reliable Across Domains Many models display inconsistencies across all cognitive levels, demonstrating an uneven development of cognitive capabilities. For example, MISTRAL-8B-IT achieves 91.62 in *Remembering* but drops sharply to 66.92 in *Applying* and 70.33 in *Creating*, reflecting surface-level fluency that does not generalize to tasks requiring flexible reasoning or creativity. In contrast, GPT-4O maintains a relatively narrow performance band, showing that sophisticated models benefit more from structured revision and are more capable of consistent reasoning across cognitive levels. Additionally, Table 2 shows that closed-source models outperform open-source ones in terms of final output quality. This may be attributed to more extensive training data and better instruction tuning, which help closed-source models generate more coherent, human-like questions.

However, illustrated in Table 2, we observe that LOT skills, e.g., *Remembering*, are easy for LLMs to perform and improve through revision. Particularly, open-source models show strong gains in this category across rounds, indicating that LLMs are highly responsive to structured feedback when dealing with rote or surface-level tasks. This pattern aligns with the characteristics of "System 1" cognition, which reflects that the THINK framework is able to isolate and evaluate effectively.

Smaller Models Are Efficient But Limited in Quality Ceiling Table 2 reveals an interesting trend in refinement efficiency. Models with smaller parameter counts, e.g., MISTRAL-8B-IT, achieve

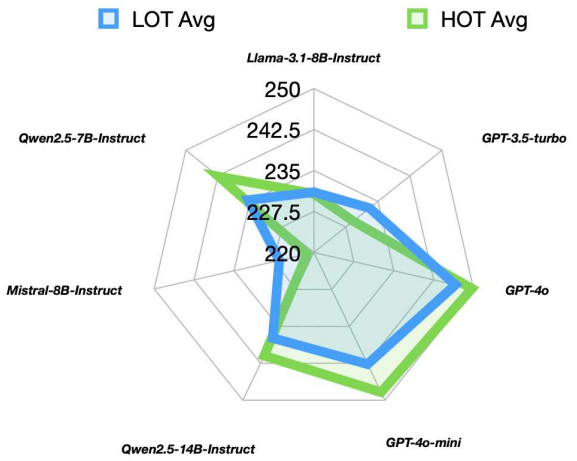

Figure 3: Comparison between HOT and LOT. The scale is the sum of scores across corresponding levels.

lower average R_{pass} values, indicating faster convergence during iterative revision. However, this efficiency often comes at the cost of lower final quality scores, reflecting a trade-off between revision speed and output quality. These findings suggest that while smaller models may adapt feedback quicker, larger models exhibit a greater capacity for sustained, high-quality refinement.

Feedback-driven Learning Enhances Higher-Order Thinking As shown in Table 1, performance in higher-order cognitive categories, including *Analyzing*, *Evaluating*, and *Creating*, often outperforms that in mid-level categories. For example, QWEN2.5-14B-IT scores above 77 in all higher-order dimensions. This suggests that the feedback-driven learning is particularly effective in improving deeper reasoning abilities that may not be captured in single-turn evaluations, highlighting its value in diagnosing higher-order cognitive competencies.

Critically, as illustrated in Figure 3, the closed-source model shows promising results in HOT

skills. This reinforces that advanced, instruction-tuned models are better positioned to engage with feedback-driven reasoning tasks. These models may have implicitly learned to perform tasks aimed at HOT during training on feedback, an advantage that becomes visible only under frameworks like THINK.

Qualitative Assessment of THINK-guided Enhancement Quality Table 3 presents a representative example evaluated across three conditions: the original flawed question, a zero-shot variant using only the "Five Keys" prompt (Appendix A.2), and the output generated by the THINK framework. Qualitative analysis of the outputs was further conducted, with detailed evaluations summarized in Table 4 and full annotations provided in Appendix C.

The comparison reveals that both zero-shot and THINK-guided responses improve over the original, but the THINK framework leads to more consistent gains in contextual reasoning and conceptual accuracy. In particular, it correctly identifies that the duration of a musical performance is invariant to ensemble size, avoiding the erroneous inverse proportionality assumption present in the original and baseline outputs. In other words, zero-shot models fail to identify inconsistencies between problem conditions and real-world environments, leading to misleading improvements in problem generation.

Moreover, the THINK-guided output engages HOT skills. While the baseline reflects misapplied procedural logic, and the zero-shot version resolves surface-level errors, the THINK response exhibits abstraction and analysis consistent with upper levels of Bloom's taxonomy. It unpacks implicit assumptions, maintains narrative plausibility, and applies structurally coherent reasoning. These results indicate that the THINK framework enhances not only accuracy but also the depth and generalizability of model reasoning.

Conclusion

In this work, we introduce THINK, a multi-agent evaluation and feedback-driven framework grounded in educational theory, to diagnose and improve higher-order thinking skills in large language models. THINK systematically generates, critiques, and revises mathematical problems aligned with Bloom's Taxonomy, allowing detailed analysis of model reasoning beyond standard accuracy metrics, enabling us to measure model performance on applying, analyzing, and creating, not just recall. Evaluation of seven LLMs reveals a persistent HOT skills gap: models perform well on lower-order tasks, but score significantly lower on practical application and concept creation. Our framework mitigates this gap via structured feedback cycles and demonstrates that closed-source models currently outperform open-source ones in reasoning quality. Qualitative analysis confirms that THINK-guided outputs exhibit deeper conceptual alignment and domain fidelity.

By making models "think-aloud" through iterative critique, THINK offers a scalable, principled approach for the community to both measure and advance LLM cognition, paving the way for more robust reasoning capabilities in educational and real-world applications. Future work could extend THINK in several promising directions, including exploring cross-domain transfer by applying our framework to other reasoning tasks beyond mathematics, and integrating THINK into human evaluation workflows to support the development of more effective human-AI collaborative reasoning systems in educational contexts.

Limitation

This study does face certain limitations as it is a preliminary framework. While our framework demonstrates strong potential, several aspects warrant further exploration. The current study relies on a curated set of flawed mathematical problems, which may limit the diversity of error types encountered in broader settings. Future work could benefit from incorporating more varied, real-world data to enhance generalizability. Additionally, although the evaluation rubric was designed to be lightweight and prompt-efficient, more comprehensive scoring frameworks could offer deeper insights into reasoning quality and consistency. At the same time, this study did not recruit external experts for output verification, which may reduce the reliability of THINK in practical applications. Finally, THINK aims to improve HOT skills performance, there is a risk that optimization toward rubric-aligned outputs could encourage overfitting to evaluative heuristics. To mitigate this, we emphasize diverse tasks and maintain transparency about rubric design. Broader adoption should be accompanied by careful validation to avoid reinforcing narrow benchmarks of "correctness" in open-ended reasoning tasks.

Ethical Considerations

This study involves the evaluation of large language models using synthetic and publicly available mathematical problem data. No personally identifiable information or human subject data were used in model evaluation.

References

Peter W Airasian. 2001. *A taxonomy for learning, teaching, and assessing: A revision of Bloom's Taxonomy of Educational Objectives*. Longman.

Maryam Amirizaniani, Elias Martin, Maryna Sivachenko, Afra Mashhadi, and Chirag Shah. 2024. Do llms exhibit human-like reasoning? evaluating theory of mind in llms for open-ended responses. *arXiv preprint arXiv:2406.05659*.

M Borge, BK Smith, and T Aldemir. 2024. Using generative ai as a simulation to support higher-order thinking. *International Journal of Computer-Supported Collaborative Learning*, 19(4):479–532.

Seong Yeub Chu, Jong Woo Kim, and Mun Yong Yi. 2025. Think together and work better: Combining humans' and llms' think-aloud outcomes for effective text evaluation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–23.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Katherine M Collins, Catherine Wong, Jiahai Feng, Megan Wei, and Joshua B Tenenbaum. 2022. Structured, flexible, and robust: benchmarking and improving large language models towards more human-like behavior in out-of-distribution reasoning tasks. *arXiv preprint arXiv:2205.05718*.

Ioan Fazey. 2010. Resilience and higher order thinking. *Ecology and Society*, 15(3).

Jennifer Haase, Paul H. P. Hanel, and Sebastian Pokutta. 2025. Has the creativity of large-language models peaked? an analysis of inter- and intra-llm variability.

Thilo Hagendorff, Sarah Fabi, and Michal Kosinski. 2023. Human-like intuitive behavior and reasoning biases emerged in large language models but disappeared in chatgpt. *Nature Computational Science*, 3(10):833–838.

Yinghui He, Yufan Wu, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 2023. Hi-tom: A benchmark for evaluating higher-order theory of mind reasoning in large language models. *arXiv preprint arXiv:2310.16755*.

Bart Holterman and Kees van Deemter. 2023. Does chatgpt have theory of mind? *arXiv preprint arXiv:2305.14020*.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, and 1 others. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *Advances in Neural Information Processing Systems*, 37:19209–19253.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. Trust or escalate: Llm judges with provable guarantees for human agreement. *arXiv preprint arXiv:2407.18370*.

Xiaoqiang Kang, Zimu Wang, Xiaobo Jin, Wei Wang, Kaizhu Huang, and Qiufeng Wang. 2025. Template-driven llm-paraphrased framework for tabular math word problem generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24303–24311.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016. A theme-rewriting approach for generating algebra word problems. *arXiv preprint arXiv:1610.06210*.

David R. Krathwohl. 2002. A revision of bloom's taxonomy: An overview. *Theory Into Practice*, 41:212 – 218.

Sotiris Lamprinidis. 2023. Llm cognitive judgements differ from human. In *International conference on frontiers of artificial intelligence, ethics, and multidisciplinary applications*, pages 17–23. Springer.

Ehsan Latif, Yifan Zhou, Shuchen Guo, Yizhu Gao, Lehong Shi, Matthew Nayaaba, Gyeonggeon Lee, Liang Zhang, Arne Bewersdorff, Luyang Fang, and 1 others. 2024. A systematic assessment of openai o1-preview for higher order thinking in education. *arXiv preprint arXiv:2410.21287*.

Llama. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Qian Niu, Junyu Liu, Ziqian Bi, Pohsun Feng, Benji Peng, Keyu Chen, Ming Li, Lawrence KQ Yan, Yichao Zhang, Caitlyn Heqi Yin, and 1 others. 2024. Large language models and cognitive science: A comprehensive review of similarities, differences, and challenges. *arXiv preprint arXiv:2409.02387*.

OpenAI. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Margus Pedaste, Mario Mäeots, Leo A Siiman, Ton De Jong, Siswa AN Van Riesen, Ellen T Kamp, Constantinos C Manoli, Zacharias C Zacharia, and Eleftheria Tsourlidaki. 2015. Phases of inquiry-based

learning: Definitions and the inquiry cycle. *Educational research review*, 14:47–61.

Oleksandr Polozov, Eleanor O'Rourke, Adam M Smith, Luke Zettlemoyer, Sumit Gulwani, and Zoran Popovic. 2015. Personalized mathematical word problem generation. In *IJCAI*, pages 381–388.

Ben Prystawski, Paul Thibodeau, Christopher Potts, and Noah D Goodman. 2022. Psychologically-informed chain-of-thought prompts for metaphor understanding in large language models. *arXiv preprint arXiv:2209.08141*.

Qwen2.5. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Farzad Radmehr and Michael Drake. 2018. An assessment-based model for exploring the solving of mathematical problems: Utilizing revised bloom's taxonomy and facets of metacognition. *Studies in Educational Evaluation*, 59:41–51.

Aya Ragab, Ahmed Kaid, and Ahmed Khamis Sayed. 2024. Enhancing higher order thinking skills (hots) in education: Strategies and outcomes. *TOFEDU: The Future of Education Journal*, 3(5):1488–1499.

Steffen Saifer. 2018. *HOT skills: Developing higher-order thinking in young learners*. Redleaf Press.

Nicy Scaria, Suma Chenna, and Deepak Subramani. 2024a. How good are modern llms in generating relevant and high-quality questions at different bloom's skill levels for indian high school social science curriculum? In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 1–10.

Nicy Scaria, Suma Dharani Chenna, and Deepak Subramani. 2024b. Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation. In *International Conference on Artificial Intelligence in Education*, pages 165–179. Springer.

Karim Shabani, Mohamad Khatib, and Saman Ebadi. 2010. Vygotsky's zone of proximal development: Instructional implications and teachers' professional development. *English language teaching*, 3(4):237–248.

Ramya Srinivasan, Hiroya Inakoshi, and Kanji Uchino. 2023. Leveraging cognitive science for testing large language models. In *2023 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 169–171. IEEE.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, and 1 others. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Alberto Testolin. 2024. Can neural networks do arithmetic? a survey on the elementary numerical skills of state-of-the-art deep learning models. *Applied Sciences*, 14(2):744.

Lev Semenovich Vygotsky and Michael Cole. 1978. *Mind in society: Development of higher psychological processes*. Harvard university press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

I Wayan Widana. 2017. Higher order thinking skills assessment (hots). *JISAE*, 3(1):32–44.

I Wayan Widana, I Parwata, and I Komang Sukendra. 2018. Higher order thinking skills assessment towards critical thinking on mathematics lesson. *International journal of social sciences and humanities*, 2(1):24–32.

Michael D Wolcott and Nikki G Lobczowski. 2021. Using cognitive interviews and think-aloud protocols to understand thought processes. *Currents in Pharmacy Teaching and Learning*, 13(2):181–188.

Xiong Xiao, Yue Li, Xiuling He, Jing Fang, Zhonghua Yan, and Chong Xie. 2025. An assessment framework of higher-order thinking skills based on fine-tuned large language models. *Expert Systems with Applications*, page 126531.

Blaise Agüera y Arcas. 2022. Do large language models understand us? *Daedalus*, 151:183–197.

Yiling Yao, Stephen Hwang, and Jinfa Cai. 2021. Preservice teachers' mathematical understanding exhibited in problem posing and problem solving. *ZDM–Mathematics Education*, 53(4):937–949.

Iris Heung Yue Yim and Jiahong Su. 2025. Artificial intelligence (ai) learning tools in k-12 education: A scoping review. *Journal of Computers in Education*, 12(1):93–131.

Yunhao Zhang, Xiaohan Zhang, Chong Li, Shaonan Wang, and Chengqing Zong. 2024. Mulcogbench: A multi-modal cognitive benchmark dataset for evaluating chinese and english computational language models. *arXiv preprint arXiv:2403.01116*.

Yunpu Zhao, Rui Zhang, Wenyi Li, Di Huang, Jiaming Guo, Shaohui Peng, Yifan Hao, Yuanbo Wen, Xingui Hu, Zidong Du, Qi Guo, Ling Li, and Yunji Chen. 2024. Assessing and understanding creativity in large language models. *ArXiv*, abs/2401.12491.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Qingyu Zhou and Danqing Huang. 2019. Towards generating math word problems from equations and topics. In *Proceedings of the 12th international conference on natural language generation*, pages 494–503.

Anat Zohar and Yehudit J Dori. 2003. Higher order thinking skills and low-achieving students: Are they mutually exclusive? *The journal of the learning sciences*, 12(2):145–181.

Appendix

A Think-aloud Structure

A.1 The Five Keys Components of MWP

We introduce a practical decomposition of cognitive rigor in math problem design, termed the "Five Keys" components. This schema is rooted in educational research on instructional design and cognitive development (Airasian, 2001; Radmehr and Drake, 2018), and is aligned with the Revised Bloom's Taxonomy to ensure both depth of knowledge and metacognitive engagement.

This decomposition provides a structured lens for evaluating whether a math problem—and by extension, an LLM's solution process—demonstrates authentic HOT skills. Rather than emphasizing rote correctness, each component targets different facets of complex reasoning, enabling a multi-dimensional assessment of LLM behavior within the THINK framework.

1. **Math Concepts and Domains**: This dimension identifies the core mathematical ideas underlying a task, such as algebraic structures, number theory, or geometry. By analyzing which concepts are invoked, we assess the knowledge dimension activated during problem-solving and whether the LLM navigates these domains coherently.

2. **Prerequisite Skills**: This component captures the foundational knowledge—both conceptual and procedural—that a learner or model must possess to attempt a solution. These skills serve as proxies for prior knowledge and inform whether the LLM draws upon relevant background competence.

3. **Mathematical Representations**: These include formal expressions (e.g., symbolic notation, equations), diagrams, or stepwise procedures. Representations are critical for logical coherence and traceability in reasoning. Evaluating this component helps identify whether an LLM applies operations in a structured and intelligible manner.

4. **Alternative Values**: This refers to variations in the input parameters of a problem that preserve its underlying structure. A model's ability to adapt its reasoning across such variants reflects generalization ability—an essential attribute of HOT.

5. **Narrative Stories**: Embedding problems in real-world or socio-cultural contexts situates abstract mathematical reasoning within meaningful scenarios. This component supports engagement and contextual transfer, and allows us to probe whether the LLM can maintain reasoning integrity when the task is couched in diverse narrative frames.

By formally integrating these components into our evaluation, we enable a principled analysis of LLM reasoning behaviors. Each element supports the dual objectives of cognitive rigor and metacognitive awareness, offering a richer and more educationally grounded alternative to traditional correctness-based metrics. The "Five Keys" thus serve as a pedagogical bridge between human-centered learning sciences and machine learning evaluation, reinforcing the interpretability and validity of the THINK framework.

A.2 Five Keys Prompt Details

Five Keys Improvement

You are a mathematical problem-maker, and at the same time an expert in cognitive science, psychology, philosophy and education. As an LLM you can generate contents related to requirements, and now your purpose is to self-reflect on the process of your math problem generation process, analyzing what you have done.

Remember, this is your problem generation outcome last time. Think aloud as you work on the instructions:

1. Analyze the generated problem of the last round. You should try to understand and retrieve the specific mathematical information in it such as facts, patterns, objects, or contextual information, and decipher these meanings.
2. Use cognitive skills essential for processing and applying information effectively. It includes understanding and organizing information, analyzing relationships, drawing conclusions, and distinguishing nuances. Additionally, you should evaluate ideas critically.
3. Generate mathematical expressions for the new problems. These new expressions should have the same form as the given expressions in the previous generated math problem. They must have the same complexity as well. Choose values to substitute into the expression, and calculate the outputs.
4. Generate stories for these mathematical expressions with the appropriate questions based on the chosen values. The generated stories must be a mathematical word problem with the corresponding expressions. The story must be creative and unique.
5. Following and combining the previous steps, and you will generate a new creative version of the given math problem. Review the generated new version math problem, ensuring all the criteria are satisfied and double check it.

Provide your evaluation in JSON format with these exact keys:
{{
"question": "The complete question text",
"solution": "The detailed solution approach"
}}

Please also address these improvement suggestions {json.dumps(improvement_suggestions, indent=2)}

B THINK framework details

B.1 Synthetic Bad-quality Question Prompt

Bad-quality Question Generator

You are an expert in creating intentionally flawed math questions. Your task is to generate a single math question that has one or more of the following issues:
1. Ambiguous wording or missing critical information
2. Unrealistic assumptions or scenarios
3. Multiple possible interpretations
4. Contradictory information
5. Unclear requirements or expectations

The question should follow this format:
{{
"ID": null,
"question": "The question text",
"LaTeX question": "The question text with LaTeX
formatting",
"solution": "Explanation of why the question is flawed and what information is missing or ambiguous",
"mathConcept1": "Main math concept (e.g., Arithmetic and Algebra)",
"mathConcept2": "Sub-concept (e.g., Algebraic expressions)",
"mathConcept3": "",
"Difficulty": "N/A or Easy/Medium/Hard",
"Grade": "9 12 or 6 8 or College",
"Resource": "GPT"
}}

Make sure the question has a clear flaw that makes it difficult to solve or has multiple valid interpretations.

B.2 Multi-Agent Evaluation Prompts - $A_1, A_2, ..., A_6$

Remembering - level 1

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Remembering" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Step 1: Identify "Big Five" Components. Extract these from both problems: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values that substitute into expressions, and 5) creative and unique narrative story based on real-life socio-cultural experiences.
Step 2: Remembering. Compare the five components in both problems. The score should represent how well the math problem generator remembers and retains critical information and components from the old problem in the new version.
Step 3: Levels of Remembering.
- Strong Remembering (80-100): If all math concepts, required skills, math expressions, and the narrative story in the new problem are almost the same as in the old problem, assign a performance_score between 80 and 100.
- Medium Remembering (60-80): If two out of the following four components are similar between the new and old problems (math concepts, required skills, math expressions, and the narrative story), assign a performance_score between 60 and 80.
- Low Remembering (<60): If less than two of these components are shared, assign a performance_score between 0 and 60. Note that the 'values' component is not considered in this step for partial similarity.
Step 4: Confidence Score and Suggestion. Reflect on your confidence level in making this judgment and assign a confidence_score between 0 and 100. Provide actionable and specific suggestions to enhance the problem as improvement_suggestions.

Details for Comparison:
- **Previous Problem:**: {last_question_details}
- **Previous Expected Solution:**{last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:**{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{ "performance_score": 0-100,
"confidence_score": 0-100 }}

Understanding - level 2

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Understanding" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Step 1: Identify "Big Five" Components. Extract these from both problems: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values that substitute into expressions, and 5) creative and unique narrative story based on real-life socio-cultural experiences.
Step 2: Understanding. Compare the five components to assess whether the generator effectively modifies the problem across seven subcategory operations: interpreting, exemplifying, classifying, summarizing, inferring, comparing, and associating.
Step 3: Levels of Understanding.
- Strong Understanding (80–100): Demonstrates a deep grasp of the five components, identifying at least three operations among the seven.
- Medium Understanding (60–80): Reflects surface-level changes, identifying at least one operation among the seven.
- Low Understanding (<60): Shows minimal variation, with errors and inconsistencies. The new problem fails to demonstrate the generator's ability across the seven operations.
Step 4: Confidence Score and Suggestion. Reflect on your confidence level in making this judgment and assign a `confidence_score` between 0 and 100. Provide actionable and specific suggestions to enhance the problem as `improvement_suggestions`.

Details for Comparison:
- **Previous Problem:**: {last_question_details}
- **Previous Expected Solution:**{last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:**{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{
"performance_score": 0-100,
"confidence_score": 0-100
}}

Applying - level 3

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Applying" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Step 1: Identify "Big Five" Components: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values used, and 5) creative narrative.
Step 2: Applying. Look for evidence that the generator applies constructed knowledge to both familiar (executing) and unfamiliar (implementing) tasks.
Step 3: Levels of Applying:
- Strong Applying (80–100): Demonstrates effective knowledge application and introduces useful variation or improvement.
- Medium Applying (60–80): Applies prior knowledge in familiar form with limited creativity.
- Low Applying (<60): Mostly replicates prior problem without deeper application.
Step 4: Confidence Score and Suggestion. Assign a `confidence_score` and suggest specific improvements.

Details for Comparison:
- **Previous Problem:**:
{last_question_details}
- **Previous Expected Solution:**
{last_question_expected_solution}
- **New Problem:**
{new_question_details}
- **New Expected Solution:**
{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{
"performance_score": 0-100,
"confidence_score": 0-100
}}

Analyzing - level 4

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Analyzing" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Please follow these steps:

Step 1: Identify "Big Five" Components: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values used, and 5) creative narrative.
Step 2: Analyzing. Look for signs that the problem generator breaks down elements, highlights distinctions, and reorganizes structure.
Step 3: Levels of Analyzing:
- Strong Analyzing (80–100): Breaks down and reorganizes structure effectively to highlight deeper relationships.
- Medium Analyzing (60–80): Identifies structure but without major transformation.
- Low Analyzing (<60): Surface-level manipulation or copy with minimal analysis.
Step 4: Confidence Score and Suggestion. Assign a `confidence_score` and suggest specific improvements.

Details for Comparison:
- **Previous Problem:**: {last_question_details}
- **Previous Expected Solution:**{last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:**{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{
"performance_score": 0-100,
"confidence_score": 0-100
}}

Evaluating - level 5

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Evaluating" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Please follow these steps:

Step 1: Identify "Big Five" Components: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values used, and 5) creative narrative.
Step 2: Evaluating. Examine whether the generator makes justified choices, defends reasoning, and prioritizes design decisions.
Step 3: Levels of Evaluating:
- Strong Evaluating (80–100): Provides justified changes and demonstrates prioritization in design logic.
- Medium Evaluating (60–80): Modifies problem with some justifications or preference reasoning.
- Low Evaluating (<60): Minor edits without clear evaluation or rationale.

Step 4: Confidence Score and Suggestion. Assign a `confidence_score` and suggest specific improvements.

Details for Comparison:
- **Previous Problem:**: {last_question_details}
- **Previous Expected Solution:**{last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:**{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{
"performance_score": 0-100,
"confidence_score": 0-100
}}

Creating - level 6

You are an expert in math and reasoning, acting as a refiner and evaluator, to assess the "Creating" level skills of a math problem generator by comparing a newly generated math problem with a previous one.

Evaluation Criteria
Please follow these steps:

Step 1: Identify "Big Five" Components: 1) math concepts and domains, 2) required skills to solve the problem, 3) math expressions as sequence of operations, 4) values used, and 5) creative narrative.
Step 2: Creating. Assess whether the generator develops original content by synthesizing and inventing meaningful structure or context.
Step 3: Levels of Creating:
- Strong Creating (80–100): Constructs novel and effective problem with well-integrated ideas.
- Medium Creating (60–80): Makes some changes or combinations with partial novelty.
- Low Creating (<60): Mostly rearranges or copies with minimal originality.

Step 4: Confidence Score and Suggestion. Assign a `confidence_score` and suggest specific improvements.

Details for Comparison:
- **Previous Problem:**: {last_question_details}
- **Previous Expected Solution:**{last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:**{new_question_expected_solution}

Result Format:
Provide your evaluation in JSON format with these exact keys:
{{
"performance_score": 0-100,
"confidence_score": 0-100
}}

B.3 Holistic Evaluation Agent - A_7

Holistic Evaluation - General Quality

You are an expert evaluator assessing Math Problem Quality and Math Language Quality in the educational question generation research context.
Please evaluate the quality of the following math word problem by analyzing its big five components and linguistic features. Identify and categorize any linguistic-level errors (e.g., ambiguity, unanswerability, or linguistic complexity) and assess the problem's solution strategy.

Details for Comparison:
- **Previous Problem:** {last_question_details}
- **Previous Expected Solution:** {last_question_expected_solution}
- **New Problem:** {new_question_details}
- **New Expected Solution:** {new_question_expected_solution}

Step 1: Big Five Components Extraction
1) Math concepts and domains
2) Required skills to solve the problem
3) Math expressions as sequence of operations
4) Values that substitute into expressions
5) The narrative story based on real-life socio-cultural experiences

Step 2: Lexical and Syntactic Complexity Analysis
- Type-Token Ratio (TTR)
- Yngve Score
- Frazier Score
- Frazier–Roark Score
- Developmental Level
- Syntactic Frequency
- Mean Dependency Distance (MDD)
- Sentence Length

Step 3: Error Identification and Classification
- Ambiguity
- Unanswerability
- Rationality

Step 4: Solution Strategy Analysis
- One-Step or Multi-Step
- Comprehension Challenges from Multi-Step Reasoning

Step 5: Improvement Suggestions
Suggestions should address:
- Ambiguous phrasing
- Unanswerable problems
- Linguistic complexity
- Structure consistency and narrative realism

Step 6: Performance Score Calculation (0–100)
1. Lexical and Syntactic Complexity
2. Error Count and Severity
3. Clarity and Solvability
4. Answerability Penalty
5. Structural Consistency and Creativity

Scoring Guidance:
- 90–100: Clear, simple, and error-free problem.
- 70–89: Minor complexity or errors that slightly impact clarity.
- 50–69: Moderate complexity and multiple identifiable issues.
- 0–49: Significant errors, ambiguity, or unanswerable conditions.

Result Format:
Please return your evaluation in the following JSON format:
{{
"performance_score": 0-100,
"confidence_score": 0-100,
"improvement_suggestions": ["suggestion1", "suggestion2"]
}}

C Human Expert Quality Evaluation

Evaluation Component	Zero-shot Qwen2.5-14B	THINK-Guided Qwen2.5-14B	Comparison & Insight
Math Concepts and Domains	Implies inverse proportionality between number of musicians and performance time, akin to shared work problems in algebra.	Recognizes invariance of musical performance duration, aligning with real-world temporal constraints rather than mathematical proportional reasoning.	The baseline activates inappropriate algebraic domain reasoning, while the instruction-tuned version correctly disengages from it, reflecting conceptual coherence.
Prerequisite Skills	Requires procedural knowledge of ratio and unit manipulation but misapplies them due to the incorrect premise.	Requires conceptual understanding of real-life constraints rather than computation.	The instruction-tuned version activates domain-appropriate prior knowledge, indicating better alignment with relevant mental schemas.
Mathematical Representations	Suggests (implicitly) a proportional formula: (120 musicians × 40 minutes) ÷ 60 = 80 minutes. No explicit expression, but logic implies computation.	No symbolic expression: relies on verbal conceptual reasoning that performance time is independent of musician count if ensemble is complete.	The baseline attempts structured reasoning but misapplies it; the tuned version avoids misleading formalism, showing better traceability and logic.
Alternative Values	Fails to generalize: if given different but equivalent values, the baseline would still apply faulty proportional logic.	Generalizes correctly: the model recognizes that performance duration is invariant under alternative numbers of musicians, assuming parts are covered.	Instruction tuning enhances generalization across input permutations that preserve the core problem structure.
Narrative Stories	Uses a formal orchestra setting but leverages it in a way that misleadingly maps to mathematical workload sharing.	Uses a school band narrative, maintaining realism while correctly situating the mathematical logic within a consistent real-world constraint.	The instruction-tuned model better integrates narrative realism and reasoning integrity, supporting engagement without conceptual distortion.
Bloom's Taxonomy Level	*Apply* (misapplied): Requires calculation, but the wrong concept leads to incorrect problem-solving.	*Understand / Analyze*: Requires unpacking implicit assumptions and applying invariant reasoning to a familiar context.	The instruction-tuned version ascends Bloom's hierarchy, requiring abstract thinking and transfer, not mechanical execution.

Table 4: Comparative analysis of baseline and instruction-tuned QWEN2.5-14B-IT models across multiple evaluation dimensions, highlighting improved contextual reasoning and domain-appropriate knowledge application.