mathematics

MDPI

Article

Exploring Large Language Models for University-Level Mathematics: A Comparative Study

Fan Yang *, Wanwan Xia and Wenting Qin

School of Physical and Mathematical Sciences, Nanjing Tech University, Nanjing 211816, China; 201910006533@njtech.edu.cn (W.X.); qwtxxxx@163.com (W.Q.)
* Correspondence: fanyang@njtech.edu.cn

Abstract

Large language models (LLMs) have shown great potential in solving complex mathematical problems, but their performance in university-level mathematics is still underexplored. This study provides a systematic evaluation of eleven state-of-the-art LLMs on five core undergraduate mathematics courses. An end-to-end automated pipeline is proposed for solution inference and answer evaluation, including a reliable ensemble evaluation scheme using seven reasoning-enabled LLMs as expert evaluators. The experimental results show that reasoning models outperform non-reasoning ones significantly, with DeepSeek-V3.2 and Kimi-K2.5 achieving average scores of 88.24 and 88.99, respectively. Under the conditions of the conducted experiment in this study, DeepSeek-V3.2 achieves the most reasonable accuracy–cost trade-off. This work reveals the strengths and limitations of modern LLMs in advanced mathematical reasoning and provides insights into their application in AI mathematical education.

Keywords: large language models; mathematical reasoning; university-level mathematics; ensemble evaluation; mathematical education

MSC: 68T20; 97U50

1. Introduction

Large language models (LLMs) have emerged as general-purpose tools across diverse fields including natural language processing [1,2], code synthesis [3,4], scientific discovery [5], and decision-making in complex scenarios [6,7]. State-of-the-art models such as GPT [8], Qwen [9], and DeepSeek [10,11] exhibit strong contextual understanding, multi-step reasoning, and scalability for real-world tasks. These advances have accelerated the adoption of LLMs in both industry and academia, especially for tasks requiring logical and symbolic reasoning.

In education, LLMs have attracted growing interest for enhancing teaching and learning [12,13]. Applications like MagicSchool.AI [14] use LLMs to generate personalized explanations, solve math word problems, and provide automated formative feedback. Empirical studies [15,16] demonstrate that step-by-step explanations from LLMs improve student learning outcomes, particularly when students first attempt problems independently. Consequently, LLMs are widely deployed in K-12 education [17], particularly for mathematics, and recent surveys [18,19] rank AI tutoring tools among the most popular educational applications.

Check for updates

Academic Editor: Francesco Calimeri

Received: 25 April 2026
Revised: 22 May 2026
Accepted: 27 May 2026
Published: 29 May 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Despite these developments, most research and applications remain focused on elementary and secondary mathematics [20–23]. Consequently, the performance of LLMs on advanced undergraduate mathematics remains underexplored. Specifically, little is known about their performance across core university mathematics courses. The lack of comprehensive, course-level evaluation makes it difficult to assess whether LLMs can reliably support learners in higher education.

To address this gap, this work proposes an automated framework for solution inference and answer evaluation using three public university-level mathematics datasets: MathOdyssey [24], U-MATH [25], and CollegeMath [26]. For systematic evaluation, problems are organized into five core undergraduate mathematics curricula to assess the performance of eleven modern LLMs across different mathematical domains. From the perspective of higher education and course-specific pedagogy, this study identifies the strengths and limitations of both reasoning-enabled and conventional LLMs for undergraduate mathematics learning.

The main contributions of this work are summarized as follows:

1. The implementation of an automated inference and evaluation pipeline for solution inference and answer assessment, enabling reproducible evaluation of LLMs on advanced mathematical tasks;
2. A curriculum-aware performance evaluation of eleven LLMs across five foundational undergraduate mathematics courses: Algebra, Calculus, Differential Equations, Precalculus, and Probability and Statistics;
3. A comparative analysis between reasoning-enabled and non-reasoning models, uncovering distinct patterns in overall accuracy and course-specific performance;
4. A systematic ensemble evaluation paradigm that leverages multiple LLMs as evaluators to verify model outputs, providing an extensible workflow for LLM-based mathematical answer assessment.

The remainder of this paper is structured as follows: Section 2 reviews related work on LLM-based mathematical problem solving and evaluation of mathematical solutions. Section 3 details the proposed automated problem-solving and evaluation framework, including dataset collection, preprocessing, model inference, and answer evaluation. Section 4 presents the experimental results covering quantitative metrics, model comparisons, evaluator analysis, and course-level case studies. Finally, Section 5 concludes the paper and discusses future directions.

2. Related Work

2.1. Large Language Models

Large language models advanced rapidly in recent years with significant improvements in reasoning, coding, and long-context understanding. The foundation of modern LLMs was laid in 2017, when Vaswani et al. [27] proposed the Transformer architecture and self-attention mechanisms, which revolutionized natural language processing (NLP) and became the core backbone of all subsequent LLMs.

Early models established the large-scale pre-training paradigm based on the Transformer architecture. A typical example is GPT-3, released by OpenAI in June 2020 with 175 billion parameters; it demonstrated the potential of large-scale pre-training but lacked strong instruction-following and reasoning abilities. GPT-3.5, released in November 2022, further improved performance by integrating Reinforcement Learning from Human Feedback (RLHF), significantly enhancing reasoning. In March 2023, GPT-4 [28] was released, greatly improving mathematical and logical reasoning with stronger contextual understanding and multi-step inference. Subsequent iterations continued to advance; GPT-4o, released in May 2024, extended capabilities to text, image, and audio inputs while greatly

improving inference speed. GPT-4o Mini, a compact multimodal variant, provides near-real-time responses at lower computational cost, making it suitable for interactive educational applications such as intelligent tutoring.

OpenAI also developed a series of models optimized for deep reasoning to handle complex scientific and mathematical tasks. In September 2024, the GPT-o1 series was launched, specifically designed for complex science, coding, and math tasks by simulating human Chain-of-Thought (CoT) reasoning. O3-Mini [29], released in January 2025, delivers faster inference and stronger reasoning at lower cost. In April 2025, OpenAI introduced O4-Mini [30], a lightweight model built on fourth-generation architecture. Compared with O3-Mini, O4-Mini adopts improved training strategies and optimized inference paths, focusing on applications requiring fast responses and low resource consumption. O4-Mini achieved a 99.5% score in the 2025 American Invitational Mathematics Examination (AIME), showing reliable performance on general and reasoning tasks.

Anthropic's Claude series has similarly advanced reasoning capabilities. Starting with Claude-1 and Claude-2 in 2023, the series evolved into the Claude-3 family (Haiku, Sonnet, and Opus) in February 2024. Claude-3.5 Haiku [31], released in October 2024, improves inference speed while maintaining reasoning performance comparable to Opus. Claude-3.5 Sonnet supports adaptive reasoning modes to dynamically optimize performance and speed based on task complexity.

Alibaba Cloud's Qwen series represents a major advance in open-source LLMs, with continuous iterations to improve reasoning and deployment flexibility. Initially launched in April 2023, the series has evolved through Qwen-VL, Qwen-2, and Qwen-2.5 [32], demonstrating strong performance on many downstream tasks and highly competitive chat models via RLHF. The latest Qwen-3 series [33], released in April 2025, introduces a thinking budget mechanism, allowing users to adaptively allocate computing resources during inference, balancing latency and performance according to task complexity. Importantly, Qwen-Plus supports adaptive control over reasoning depth and computational cost. These innovations enable strong performance on academic benchmarks while supporting flexible deployment in resource-constrained environments. Released in February 2026, Qwen3.5-Plus fuses linear attention with a sparse Mixture-of-Experts (MoE) structure, delivering excellent performance in reasoning, coding, agent capabilities, and multimodal understanding.

Founded in 2023, Moonshot.AI developed the Kimi series, which emphasizes ultra-long context windows and efficient document-level reasoning, establishing itself as a leader in long-context LLM research. Kimi-K1 was released in December 2024 as a visual reasoning model, and Kimi-K2 was launched in 2024. In February 2026, Kimi-K2.5 [34] was released, integrating techniques including joint text-vision pre-training, zero-vision supervised fine-tuning, and joint text-vision reinforcement learning. The model supports complex multi-step workflows, combining long-context understanding with tool use, code generation, and environmental interaction.

In parallel, DeepSeek has become a competitive provider of high-performance LLMs, focusing on reasoning and domain-specific capabilities. After releasing DeepSeek-Coder and DeepSeek-V3 in 2024 [35], the company launched DeepSeek-R1 in January 2025, a reasoning-focused model strong in symbolic logic, multi-step reasoning, and domain-specific knowledge [11]. Like other reasoning-oriented models, DeepSeek-R1 optimizes multi-step inference through RLHF and structured reasoning strategies. It shows competitive performance on mathematical and scientific benchmarks while offering advantages in efficiency and cost. In December 2025, DeepSeek developed the open-weight model DeepSeek-V3.2 [36], designed to balance computational efficiency, advanced reasoning, and agent capabilities. The model introduces DeepSeek Sparse Attention (DSA) for efficient

long-context processing and a scalable reinforcement learning framework that significantly improves reasoning performance.

2.2. Mathematical Problem Solving

A growing body of work has explored LLMs for mathematical reasoning, consistent with their increasing use in educational scenarios. Mathematical problem-solving has become a fundamental benchmark for evaluating LLMs [37], as it comprehensively tests logical reasoning, symbolic manipulation, and multi-step inference. Over the past few years, various curated datasets [20–23,37] have been developed to systematically assess LLMs on arithmetic reasoning, symbolic manipulation, and algebraic transformation, laying a foundation for evaluating mathematical capabilities.

Several widely used benchmarks target grade-school and high-school mathematics. For instance, GSM8K [20] includes 8500 high-quality elementary math word problems that require multi-step arithmetic reasoning. Its simple language and structured solutions make it a standard for evaluating step-by-step logical inference. GSMIC [38] extends GSM8K by adding extraneous information, showing that LLM performance drops significantly under distraction. Similarly, GSM-Symbolic [39] provides a symbolic template of GSM8K where values and names can be programmatically modified; it observes severe data contamination across mainstream LLMs, revealing over-reliance on training patterns rather than genuine reasoning.

The MATH [21] dataset is a widely recognized benchmark for machine learning problem-solving, consisting of 12,500 high-school competition problems covering Algebra, Geometry, and Calculus. In 2024, Functional MATH [40] was proposed to convert MATH problems into functional variants, further testing LLMs' ability to adapt to different representations while preserving core reasoning. OlympiadBench [22] is an Olympiad-level bilingual multimodal benchmark with 8476 math and physics competition problems, including Chinese national college entrance exam questions, pushing LLM evaluation toward more complex competition-style reasoning. MATH-Perturb [23] applies two types of perturbations to MATH via minimal editing: simple perturbations that preserve solution logic, and hard perturbations that fundamentally change it. This design helps distinguish genuine reasoning from pattern matching.

Despite these rich benchmarks, most existing studies focus on elementary-, high-school-, or competition-level mathematics, including arithmetic reasoning, basic algebra, and contest-style tasks. In contrast, university-level mathematics poses unique challenges, requiring more complex logical reasoning, continuous structures, and rigorous multi-step derivations that go beyond pre-university benchmarks. In recent years, some works have attempted to expand evaluation to university-level mathematics. OCWCourses [41], collected in 2022 from MIT curricula, was an early attempt to include university math problems. However, these datasets are small and still focus on Olympiad-style problems rather than practical coursework topics. In 2024, MathOdyssey [24] collected 387 expert-generated problems spanning high-school, university, and Olympiad levels; however, only 101 are university-level, a relatively small proportion. In 2025, U-MATH [25] was proposed as a university-level benchmark, with problems collected from real undergraduate coursework and equipped with correct answers, closely reflecting real academic standards. Nevertheless, most U-MATH problems focus on Algebra and Calculus, lacking coverage of other core undergraduate courses. To ensure diversity, CollegeMath [26] was collected from nine college mathematics textbooks, containing 2428 text-format records covering seven key mathematical disciplines. These records were used to train various early LLMs before 2024, making them valuable for evaluating generalization across undergraduate math.

A key limitation of the existing studies is that most evaluate LLMs on a single domain or an unstructured mix of problems, making it difficult to assess generalization across core undergraduate courses. This limitation highlights the need for a curriculum-aware evaluation framework targeting core undergraduate subjects to assess whether LLMs can meaningfully and systematically support higher education learners.

2.3. Evaluation of Mathematical Solutions

Evaluating the correctness of mathematical solutions has become an important and challenging research direction [42–45]. Unlike natural language generation, mathematical problem-solving requires strict logical rigor, symbolic accuracy, and recognition of mathematical equivalence, making evaluation far more complex than general text assessment.

Traditional evaluation methods primarily rely on exact-match accuracy, which directly compares model outputs to predefined golden answers [42,46]. However, a major limitation is that mathematical solutions often have multiple equivalent forms, and recognizing these equivalences requires expert-level knowledge and experienced personnel [47]. Such expert-dependent evaluation is labor-intensive, time-consuming, and prone to subjective bias: even experienced experts may disagree on complex or non-standard solutions, especially in advanced mathematics. These limitations restrict the practical use of traditional methods in large-scale assessment, particularly in education scenarios where AI supports teaching and personalized feedback [48].

To address these limitations, recent studies [43–45] have explored LLM-based evaluation of generated answers. Baral [49] explored using LLMs to evaluate student responses and provide automated feedback for middle-school math, showing that LLMs can reliably replace human evaluators for basic tasks. Lee et al. [50] proposed a GPT-4-based automated assessment system for primary-school open-ended math problems; their results show high consistency between human and GPT-4 judgments in most cases, though some mathematical content still presents challenges. Xia et al. [51] introduced ReasonEval, an LLM judge framework that uses validity and redundancy to characterize reasoning quality. This framework uses high-performance LLMs (e.g., GPT-4) to evaluate solutions, focusing not only on final-answer correctness but also on the validity of intermediate steps. Additionally, MathEval [52] trained a DeepSeek-7B-based answer comparison model, enabling automated validation for annually updated Chinese national college entrance exam problems.

Despite these advances, critical limitations remain in existing LLM-based evaluation methods. Many studies focus on single-task or single-domain evaluation without addressing the complexity and diversity of university-level mathematics. Furthermore, single LLM judges have inherent biases that reduce reliability, even for general reasoning tasks [53]. Evaluation stability varies significantly based on the evaluator's reasoning ability, prompt sensitivity, and internal biases, undermining result consistency. Systematic studies on automated, scalable, equivalence-aware evaluation frameworks for advanced mathematical tasks remain limited.

3. Methodology

This section introduces an end-to-end framework to systematically analyze LLM mathematical problem-solving ability across diverse university-level curricula. To ensure generalizability and educational relevance, experimental data are sourced from three public mathematical benchmarks and further categorized into five core undergraduate courses: Algebra, Calculus, Differential Equations, Precalculus, and Probability and Statistics. The framework integrates automated question parsing, LLM inference, and answer evaluation into a unified, modular pipeline covering the full workflow from input to quantitative

performance analysis. Several metrics are used to provide clear insights into the suitability of different LLMs for higher education.

To guide empirical analysis, the pipeline addresses three core research questions:

RQ1: *How do state-of-the-art LLMs perform across core undergraduate mathematics courses?*

RQ2: *What performance differences exist between reasoning-enabled LLMs and conventional non-reasoning LLMs on university-level mathematical problem-solving?*

RQ3: *How effective is the ensemble evaluation paradigm using multiple LLMs as evaluators?*

3.1. Framework Overview

The solution inference and assessment framework is shown in Figure 1. It consists of four sequential modules: Dataset Collection, Data Preprocessing, Model Inference, and Answer Evaluation. This modular design enables reproducible, systematic evaluation of LLMs on university-level mathematical problems.

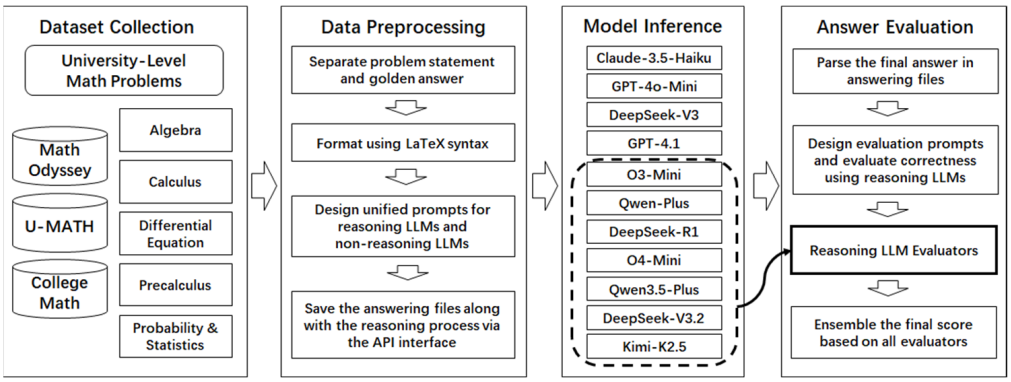


Figure 1. Overview of the proposed framework.

The Dataset Collection module collects representative problems from three public benchmarks: MathOdyssey [24], U-MATH [25], and CollegeMath [26]. All problems are categorized into five standard undergraduate curriculum categories to enable consistent course-level performance analysis.

In Data Preprocessing, raw problems are cleaned and formatted into a unified input structure containing the problem statement and corresponding golden answer. A key component is standardized prompt templates, which guide LLMs to produce clear step-by-step solutions with consistent output formats across all models. Detailed prompt settings are described in Section 3.4.

The Model Inference module sends each preprocessed problem to selected LLMs via official APIs. Both final answers and intermediate reasoning steps are collected and stored when generated. This study evaluates eleven modern LLMs and categorizes them into reasoning-enabled and non-reasoning groups based on their reasoning-oriented optimization objectives.

Finally, the Answer Evaluation module implements an ensemble evaluation mechanism. Instead of a single evaluator, multiple high-performance LLMs assess output correctness. The module automatically parses responses, verifies answers against golden solutions, and combines scores from individual evaluators to produce a robust ensemble score. Evaluation prompts and the ensemble scoring strategy are detailed in Section 3.5.

By integrating these four stages, the framework achieves strong reproducibility and scalability, providing a practical and reliable pipeline for future research on LLM performance in advanced mathematical problem-solving.

3.2. Dataset Collection

To ensure comprehensiveness and representativeness, this study selects three public university-level mathematics datasets: MathOdyssey, U-MATH, and CollegeMath. These datasets cover a wide range of undergraduate topics and provide well-defined problems with standardized answers, making them suitable for evaluating LLM performance across curricula. This section introduces each dataset, the mapping between original course categories and the five unified courses, and the quantity distribution after merging.

MathOdyssey is a curated collection of 387 expert-generated problems covering high-school, university, and Olympiad levels. Of these, 101 are university-level, spanning Calculus Analysis, Differential Equations, Linear Algebra and Abstract Algebra, Probability, and Statistics. Each problem includes a detailed solution and is categorized by difficulty, subject, and answer type, providing detailed test cases for evaluating advanced problem-solving ability.

U-MATH is a typical university-level benchmark containing 1100 undisclosed problems from U.S. undergraduate curricula, 80% of which are text-only. Its original categories cover seven core topics: Algebra, Differential Calculus, Integral Calculus, Multivariable Calculus, Precalculus, and Sequences and Series. All the problems align with undergraduate teaching standards, with clear logic and golden answers, effectively testing basic undergraduate mathematical ability.

CollegeMath is a comprehensive collection of nine college mathematics textbooks, with 2428 text-format records. Each record corresponds to a LaTeX-formatted golden answer for a math question. The textbooks cover seven key disciplines: Algebra, Precalculus, Calculus, Vector Calculus, Probability, Linear Algebra, and Differential Equations.

Because the three datasets use different original category systems, the unified five core undergraduate courses are defined in this study (Algebra, Calculus, Differential Equations, Precalculus, and Probability and Statistics) to support consistent course-level analysis. The mapping between original categories and unified ones is shown in Table 1.

Table 1. Mapping between original course categories and unified curriculum categories.

Dataset Name	Original Course Categories	Number of Problems	Unified Curriculum Categories
MathOdyssey	Calculus Analysis	24	Calculus
	Differential Equations	14	Differential Equations
	Linear Algebra and Abstract Algebra	25	Algebra
	Probability	21	Probability and Statistics
	Statistics	17	Probability and Statistics
U-MATH	Algebra	150	Algebra
	Differential Calculus	150	Calculus
	Integral Calculus	150	Calculus
	Multivariable Calculus	150	Calculus
	Precalculus	150	Precalculus
CollegeMath	Algebra	500	Algebra
	Precalculus	580	Precalculus
	Calculus	530	Calculus
	Vector Calculus	110	Calculus
	Probability	139	Probability and Statistics
	Linear Algebra	260	Algebra
	Differential Equations	309	Differential Equations

After merging, the quantity distribution in each unified category is counted and shown in Table 2, including the number of problems per category and their proportion of the total.

Table 2. Quantity distribution of problems in unified curriculum categories.

Curriculum Category	MathOdyssey	U-MATH	CollegeMath	Total	Proportion
Algebra	25	150	760	935	27.2%
Calculus	24	600	640	1264	36.9%
Differential Equations	14	0	309	323	9.4%
Precalculus	0	150	580	730	21.3%
Probability and Statistics	38	0	139	177	5.2%
Total Curricula	101	900	2428	3429	100%

The total number of problems after category unification is 3429. Calculus accounts for the largest share (36.9%), consistent with its role as a core undergraduate course. Algebra is second (27.2%), followed by Precalculus (21.3%), Differential Equations (9.4%), and Probability and Statistics (5.2%), due to fewer university-level problems in the original datasets. The dataset distribution across courses is uneven, with Calculus accounting for a substantially larger proportion of problems than Probability and Statistics. This imbalance may influence the aggregate performance metrics. To mitigate this issue, course-level analyses were conducted independently, and the overall performance was computed using macro-average curriculum accuracy rather than direct aggregation over all the problems. The detailed formulas are defined in Section 4.1.1.

3.3. Data Preprocessing

After dataset collection and category unification, data preprocessing ensures the quality, consistency, and usability of data for LLM inference and evaluation. This section details core preprocessing steps: data structure unification, text format standardization, and prompt design. The goal is to unify input formats and build a standardized prompt strategy that works for both reasoning-enabled and non-reasoning LLMs, laying a foundation for fair and reproducible evaluation.

3.3.1. Data Standardization

Due to format differences across the three datasets, data structure unification is necessary to ensure consistent LLM input and automated output parsing. Specifically, CollegeMath is originally stored in JSON, while U-MATH and MathOdyssey are CSV files from the open-source Hugging Face platform. For consistent processing under a unified API framework, each problem is split into two components: problem statement and ground-truth answer, stored separately with distinct extensions (*.problem_statement and *.golden_answer).

All the problem statements and golden answers are converted to plain text, replacing special symbols and non-standard characters with standard equivalents to avoid parsing errors. All the mathematical symbols, formulas, and equations are uniformly rendered in LaTeX, ensuring accurate recognition by LLMs, as LaTeX is widely supported for mathematical expressions in academia and AI research. Each preprocessed problem is organized into a fixed structure, including problem statement and golden answer as uniform inputs to all the LLMs. This consistent structure stabilizes model understanding and ensures that all the models receive identical information.

Since the three datasets (e.g., CollegeMath) are publicly available, potential overlap with model pretraining corpora cannot be completely excluded. To reduce potential memorization effects, metadata and dataset source information were excluded during problem-solving interactions whenever possible. In addition, the evaluation emphasizes cross-course comparative analysis rather than relying solely on absolute accuracy scores. Therefore, the reported results should be interpreted as comparative performance observa-

tions under the current setting and experimental datasets rather than absolute indicators of mathematical reasoning capability.

3.3.2. Prompt Design

Prompt design is critical, as it directly affects the quality and consistency of LLM outputs and thus the reliability of subsequent evaluation. Given capability differences between reasoning-enabled and non-reasoning LLMs, the goal is to build a unified prompt that elicits competitive performance from both groups.

The unified prompt follows a logical sequence to guide structured problem-solving. The full procedure is outlined in pseudocode function DesignUnifiedPrompt() in Algorithm 1:

Algorithm 1: Pseudocode for Unified Prompt Design Algorithm

```
Function DesignUnifiedPrompt()
Output:
  UnifiedPrompt—Unified prompt for all models
begin
  Step 1. // Role assignment strategy
          RoleDescription ← DefineRoleAssignment()
  Step 2. TaskRequirement ← DefineTask()
  Step 3. ReasoningGuidance ←AddReasoningGuidance()
  Step 4. TutorDirection ←AddTutorDirection()
  Step 5. // Standardized output requirement
          OutputFormat ← StandardizeOutput()
  Step 6. // Merge all modules into a unified prompt
          UnifiedPrompt ← Concatenate(RoleDescription,
                                      TaskRequirement,
                                      ReasoningGuidance,
                                      TutorDirection,
                                      OutputFormat)
  return UnifiedPrompt
end
```

In Step 1, the prompt defines the LLM as a university mathematics instructor expert in the five core courses. This role definition is especially important for non-reasoning models, providing explicit context to reduce ambiguity and guide task-relevant outputs. The design draws on prompt engineering research [54,55], showing that specific expert roles improve output relevance and task focus.

The second module (Step 2) clearly states core task requirements, ensuring that all the LLMs understand the scope and expectations. For example: *"Assume you are a highly experienced university mathematics professor, well-versed in advanced mathematics, linear algebra, differential equations, statistics, and probability theory. You are tasked with solving the following university-level mathematics problem accurately and clearly."*

In Step 3, reasoning guidance is embedded to support multi-step thinking, especially beneficial for reasoning models. This module provides flexible adaptive guidance compatible with both model types, aligning with the CoT literature [56], showing that even minimal guidance improves performance and interpretability on multi-step tasks. A typical instruction: *"Please reason through the problem step by step and explain all key steps, ensuring clarity and adherence to mathematical principles."*

To simulate a classroom environment and improve clarity, a tutor-mode directive is added (Step 4), encouraging an educational tone and step-by-step explanation for student comprehension: *"As a professor, guide the student with a pedagogical tone. Break down the solution clearly and ensure that the student understands both the underlying concepts and the reasoning process."*

The final module (Step 5) enforces a strict output format to enable automated parsing and evaluation. It requires the output to end with a clearly marked "Final Answers" section, with the final results in boxed LaTeX. This standardization ensures that the final answers can be automatically extracted and compared to golden answers. Extraneous text after "Final Answers" is prohibited to streamline automated evaluation.

The complete unified prompt, integrating all five modules, is shown below. It is applied uniformly to all eleven LLMs, ensuring consistent interaction and output format across inference:

"Assume you are a highly experienced university mathematics professor, well-versed in advanced mathematics, linear algebra, differential equations, statistics, and probability theory. You are tasked with solving the following university-level mathematics problem accurately and clearly. Please reason through the problem step by step and explain all key steps, ensuring clarity and adherence to mathematical principles. As a professor, guide the student with a pedagogical tone. Break down the solution clearly and ensure that the student understands both the underlying concepts and the reasoning process. Use LaTeX to format all mathematical expressions. At the end of your response, write the final answers in order for all problems. Do not include any additional text after the final answers. You must use the exact keyword ('Final Answers') with the following format:

*"Final Answers:\n" + *
*"1. \\boxed{...}\n" + *
*"2. \\boxed{...}\n" + *
"...\n""

3.4. Model Inference

Following preprocessing, eleven state-of-the-art LLMs are used to generate solutions for standardized university-level problems during model inference. The models are selected based on the following criteria to ensure a representative, feasible, and valid evaluation:

- Representativeness: All the models are recently released by major developers (OpenAI, Anthropic, Alibaba Cloud, DeepSeek, MoonshotAI), representing the state-of-the-art in research and industry, covering diverse technical routes and positioning.
- Stable API Access: All provide stable, programmable APIs, enabling large-scale batch inference and unified response collection under consistent settings.
- Architectural and Objective Diversity: The models cover diverse architectural designs and training objectives with clear reasoning ability differences, providing meaningful variation for comparative analysis.
- Demonstrated Math Competence: All the models show competitive performance in prior benchmarks involving logic, coding, or mathematics, ensuring suitability for undergraduate problems.

Based on inherent reasoning ability, the eleven LLMs are divided into two groups for comparative analysis. Reasoning-enabled LLMs refer to models explicitly optimized for multi-step reasoning through reinforcement learning and CoT training [57,58]. In contrast, non-reasoning LLMs refer to general-purpose autoregressive models without explicit reasoning-oriented optimization mechanisms. The non-reasoning group includes Claude-3.5-Haiku, GPT-4o-Mini, DeepSeek-V3, and GPT-4.1, while the reasoning-enabled group includes O3-Mini, Qwen-Plus, DeepSeek-R1, O4-Mini, Qwen3.5-Plus, DeepSeek-V3.2, and Kimi-K2.5.

The following model descriptions are summarized from publicly available technical reports, API documentation, and official model introductions provided by the corresponding organizations.

- GPT-4o-Mini (OpenAI): Lightweight variant of GPT-4o optimized for efficient inference and general-purpose text generation. The model supports instruction following and question-answering tasks without explicitly exposing dedicated reasoning procedures during inference [41].
- Claude 3.5 Haiku (Anthropic): Lightweight general-purpose LLM designed for efficient text generation and structured natural language processing tasks [31].
- DeepSeek-V3 (DeepSeek): General-purpose LLM supporting text generation, coding, and mathematical question-answering tasks based on MoE architecture [35].
- GPT-4.1 (OpenAI): General-purpose proprietary LLM supporting coding, instruction-following, and multimodal understanding tasks.
- O3-Mini (OpenAI): Lightweight reasoning-enabled model designed for multi-step reasoning and complex mathematical problem-solving tasks [29].
- Qwen-Plus (Alibaba Cloud): Reasoning-enabled LLM with multilingual capability and support for symbolic and mathematical reasoning tasks.
- DeepSeek-R1 (DeepSeek): Reasoning-enabled LLM designed for complex reasoning and mathematical problem-solving tasks. The model supports multi-step reasoning behavior for complex problems [11].
- O4-Mini (OpenAI): Compact reasoning-oriented model designed for efficient multi-step reasoning and mathematical problem-solving tasks [30].
- Qwen3.5-Plus (Alibaba Cloud): Reasoning-enabled model evolving from text-only to native multimodal and supporting multimodal understanding and mathematical reasoning tasks through MoE architecture.
- DeepSeek-V3.2 (DeepSeek): Reasoning-enabled extension of DeepSeek-V3 with additional optimization through MoE architecture for multi-step reasoning and mathematical problem solving [36].
- Kimi-K2.5 (MoonshotAI): Reasoning-enabled LLM designed for complex logical reasoning and math problem-solving tasks [34].

3.5. Answering Evaluation

3.5.1. Evaluation Prompts

After model inference, all the generated responses are evaluated using high-performance LLMs to judge correctness. The evaluation prompts are designed to guide LLMs to act as evaluators, accurately judging correctness. Similar to the unified prompt in Section 3.3.2, the evaluation prompts use a modular structure. The full procedure is outlined in pseudocode function DesignEvaluationPrompt() in Algorithm 2.

In Step 1, Evaluator Role Assignment defines the model as an expert mathematics evaluator, distinct from the problem-solving role in inference. The Evaluation Task module (Step 2) clarifies the core task: compare each LLM's final answer with the golden answer and grade according to clear rules. The specific role and task description: *"You are an expert mathematics evaluator. You will be shown a university-level math problem, the correct golden answer, and the final answers given by several AI models. Your task is to compare each model's final answer with the golden answer, and grade it accordingly."*

In Step 3, Evaluation Scoring Rules define clear, standardized criteria to guide grading and ensure consistency. Typical instructions:

"- If the model's answer is mathematically equivalent to the golden answer, mark it as correct.\n"

"- Give 100 points if the answer is correct, otherwise 0.\n"

"- Do not penalize the model for formatting or notation differences as long as the math is correct and unambiguous.\n"

Algorithm 2: Pseudocode for Evaluation Prompt Design Algorithm

```
Function DesignEvaluationPrompt ()
Output:
  EvaluationPrompt—Evaluation prompt for evaluator models
begin
  Step 1. // Role assignment for evaluator
          EvaluatorRole ←DefineEvaluatorRole()
  Step 2. EvaluationTask ← DefineEvaluationTask()
  Step 3. ScoreRules ← DefineScoringRules()
  Step 4. // Standardized output requirement
          EvaluationOutputFormat ← StandardizeEvaluationOutput()
  Step 5. // Merge all modules into an evaluation prompt
        EvaluationPrompt ← Concatenate(EvaluatorRole,
                                       EvaluationTask,
                                       ScoreRules,
                                       EvaluationOutputFormat)
  return EvaluationPrompt
end
```

The next module, Step 4, enforces a fixed output format for automated parsing and statistical analysis, matching the scoring rules. The output format is explicitly specified:

"Please output your evaluation using the following exact format: \n\n"
"Evaluation Results: \n"
"Model 1: score = 100, correct = 1/1\n"
"Model 2: score = 0, correct = 0/1\n"
"...\n\n"
"Strict output instructions: \n"
"- Use exactly the phrase 'Evaluation Results:' as the heading (case-sensitive).\n"
"- Do not include any additional comments, summaries, or explanation.\n"
"- Do not use any other headings or phrases.\n"
"- Do not add any text after the final model evaluation line.\n"

The complete unified evaluation prompt is applied uniformly to all the evaluator models. Since model identities are anonymized (e.g., "Model 1" or "Model 2"), evaluators cannot infer origin, avoiding bias and laying a foundation for a subsequent expert-voting ensemble scoring scheme.

3.5.2. Ensemble Scoring Scheme

To further improve evaluation reliability and objectivity, this study proposes an ensemble scoring scheme based on expert voting. The core idea is to use seven selected high-performance reasoning-enabled evaluators (each as an expert) to independently assess each response and then use majority voting to determine the final result and score. This approach follows expert voting logic, avoids one-sidedness from single evaluators, and reduces the impact of random errors on LLM-based evaluation.

The previous seven reasoning-enabled LLMs are selected as evaluators: O3-Mini, Qwen-Plus, DeepSeek-R1, O4-Mini, Qwen3.5-Plus, DeepSeek-V3.2, and Kimi-K2.5. These models are chosen for their strong reasoning ability and reliable performance.

Consistent with expert voting, the final result and score for each response are determined by the majority vote of the seven evaluators. For example, if 4 of 7 evaluators mark Correct (100 points) and 3 mark Incorrect (0 points), the final result is Correct with a score of 100. This voting strategy effectively reduces individual evaluator bias and random errors, ensuring objective and reliable final scores. Such a design helps enhance the stability and generalizability of evaluation outcomes in real-world educational assessment scenarios.

To further evaluate the reliability of those evaluators and validate the effectiveness of the ensemble scheme, detailed quantitative analysis and ablation experiments are presented in Sections 4.2.4 and 4.2.7, respectively.

4. Experiments

4.1. Evaluation Metric

4.1.1. Evaluation Metrics for Tested LLMs

To evaluate the performance of LLMs in problem solving, several quantitative metrics are used, focusing on cross-course differences and overall performance. For each of the five core undergraduate curricula, the course-level accuracy score is the average score of the LLM's responses in that course. Let c be a specific curriculum, N_c be the number of responses in curriculum c, and $S_{i,c} \in [0, 100]$ bethe score of the i-th response in c:

$$CourseLevel_AccScore_c = \frac{1}{N_c} \sum_{i=1}^{N_c} S_{i,c} \tag{1}$$

This metric analyzes performance across curricula and identifies strengths and weaknesses in specific mathematical domains.

The overall accuracy score is defined as the macro-average accuracy across all curricula for a single LLM. Higher scores indicate stronger overall problem-solving performance across different mathematical courses. Let C denote the total number of curricula and $CourseLevel_AccScore_c$ denote the course-level accuracy score for curriculum c. The overall performance of the m-th LLM is defined as

$$Overall_AccScore_m = \frac{1}{C} \sum_{c=1}^{C} CourseLevel_AccScore_c \tag{2}$$

To compare the overall performance gap between non-reasoning and reasoning-enabled groups, the group average score is calculated for each group. Let g be the model group (non-reasoning or reasoning-enabled), M_g the number of LLMs in group g, and $Overall_AccScore_m$ the overall accuracy score of the m-th LLM in group g:

$$Group_AvgScore_g = \frac{1}{M_g} \sum_{m=1}^{M_g} Overall_AccScore_m \tag{3}$$

4.1.2. Evaluation Metrics for Evaluator Models

For evaluator models, two metrics are adopted to assess their consistency and agreement with the final ensemble result. These metrics reflect how well a single evaluator's grading aligns with the consensus of all seven, directly indicating individual evaluator reliability.

The first metric is agreement rate, measuring consistency between each evaluator and the final ensemble score from expert voting. Let e be a specific evaluator, A_e the number of samples where the evaluator's result matches the ensemble score, and T the total number of evaluated samples:

$$AgreementRate_e = \frac{A_e}{T} \tag{4}$$

Higher values indicate greater alignment with expert consensus and more reliable grading. The second metric is inter-evaluator consistency, using the Kappa coefficient to measure grading consistency among the seven evaluators:

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{5}$$

where P_o is the observed agreement proportion and P_e is the expected agreement by chance. A high Kappa coefficient ensures that the expert voting strategy is effective and the results are reliable.

To assess the statistical robustness of the evaluation results, 95% confidence intervals (CIs) are computed for the overall accuracy score, agreement rate, and Kappa coefficient using bootstrap resampling. These intervals provide an estimate of the uncertainty associated with each metric and help determine whether the observed performance differences are likely attributable to random variation. Specifically, bootstrap resampling with 1000 iterations is performed on the evaluation samples to construct the corresponding confidence intervals for each metric. The 2.5th and 97.5th percentiles of the bootstrap distributions are used as the lower and upper CI bounds, respectively.

4.2. Experimental Results

This section reports the experimental results, including experimental setup, overall LLM performance, course-level comparison, evaluator reliability analysis, cost comparison, and a case study of special phenomena.

4.2.1. Experimental Setup

This study defines a unified API framework to standardize interactions with each selected LLM. All the experiments were conducted from March 2025 to April 2026 using publicly accessible commercial APIs provided by the corresponding model organizations or official cloud platforms like Alibaba Cloud. The API source code was implemented in Python 3.8.0 using the OpenAI API version 1.63.2. Temperature, max token limits, and random seed settings followed the default configurations of the corresponding APIs unless otherwise specified. For reasoning-enabled models, reasoning or thinking modes were enabled when supported by the corresponding APIs. During API interaction, several error conditions were handled, including server overload, service unavailability, and maximum token length exceptions. Failed requests were automatically retried after a fixed delay. The maximum number of retries was set to 10, with a retry interval of 30 s.

In the inference module, response generation of each math problem is structured as follows: each model generates a raw answer file (*.answer); for models supporting explicit reasoning, intermediate reasoning traces are stored in a separate reasoning file (*.reasoning). From these outputs, the final predicted answers are automatically extracted using predefined parsing rules and consolidated into a solution file (*.solution). Both groups of LLMs are evaluated within a consistent pipeline.

Subsequently, the answer evaluation module assesses all the solution files against corresponding golden references (*.golden_answer), resulting in a structured evaluation file (*.eval_answer). A post-processing script is programmed to parse these files and get the evaluation results using the defined metrics.

To ensure reproducibility of this study, prompts, evaluation codes, representative raw outputs, evaluation results, experimental configuration examples and Subset100 results are included in the repository: https://github.com/fftechnj/umath-problem-llm-evaluation (accessed on 13 May 2026).

4.2.2. Model Performance

This section focuses on the overall performance of the eleven tested LLMs using Formula (2), emphasizing the performance differences between the non-reasoning and reasoning-enabled groups. The overall accuracy scores and corresponding 95% confidence intervals (CIs) are summarized in Table 3.

Table 3. Overall accuracy scores of eleven tested LLMs.

Model Name	Overall Accuracy Score	95% CI	Model Group
Claude-3.5-Haiku	47.74	[45.67, 49.70]	Non-Reasoning
GPT-4o-Mini	56.76	[54.84, 58.79]	Non-Reasoning
DeepSeek-V3	76.59	[74.47, 78.47]	Non-Reasoning
GPT-4.1	82.95	[81.29, 84.51]	Non-Reasoning
O3-Mini	85.59	[83.87, 87.12]	Reasoning-Enabled
Qwen-Plus	85.59	[83.86, 87.09]	Reasoning-Enabled
DeepSeek-R1	86.12	[84.43, 87.71]	Reasoning-Enabled
O4-Mini	86.71	[85.26, 88.20]	Reasoning-Enabled
Qwen3.5-Plus	88.10	[86.63, 89.43]	Reasoning-Enabled
DeepSeek-V3.2	88.24	[86.70, 89.68]	Reasoning-Enabled
Kimi-K2.5	88.99	[87.53, 90.36]	Reasoning-Enabled

To better visualize performance differences, Figure 2 shows a bar chart of the overall accuracy scores with specific values marked above each bar. This visualization clearly illustrates the increasing performance trend from non-reasoning LLMs to reasoning-enabled LLMs.

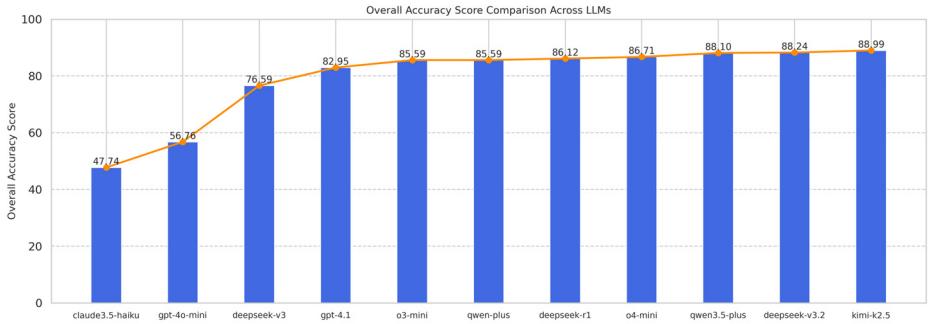


Figure 2. Overall accuracy score comparison across LLMs.

As shown in Table 3 and Figure 2, the overall performance varies widely across the eleven LLMs. Scores range from 47.74 (Claude-3.5-Haiku) to 88.99 (Kimi-K2.5). Among the reasoning-enabled models, Qwen3.5-Plus, DeepSeek-V3.2, and Kimi-K2.5 achieve the highest overall accuracy scores of 88.10, 88.24, and 88.99, respectively. However, their confidence intervals partially overlap, indicating that the performance differences among these top-performing models remain relatively small. The bootstrap confidence intervals further demonstrate that the evaluation results are statistically stable, with all the top-performing models exhibiting relatively narrow uncertainty ranges.

In contrast, earlier lightweight models such as Claude-3.5-Haiku and GPT-4o-Mini achieve much lower scores (47.74 and 56.76), both below 60. This indicates that these models lack sufficient reasoning ability for complex or multi-step mathematical tasks. Among the non-reasoning models, DeepSeek-V3 and GPT-4.1 perform relatively well, with scores of 76.59 and 82.95 respectively. This shows these models can solve many problems effectively but are less stable than the top reasoning-enabled models.

To quantitatively analyze the performance gap between groups, group average scores are calculated using Formula (3), with the results shown in Table 4.

Table 4. Group average score of the two groups.

Group Type	Non-Reasoning Group	Reasoning-Enabled Group	Score Gap
Group Average Score	66.01	87.05	21.04

The reasoning-enabled group averages 21.04 points higher than the non-reasoning group. This further demonstrates the advantages of the reasoning-enabled models under the evaluated mathematical tasks.

Additionally, the non-reasoning group shows large internal performance variation. GPT-4.1 scores 82.95, 35.21 points higher than Claude-3.5-Haiku (47.74). This indicates widely varying problem-solving abilities among the non-reasoning models. In contrast, the reasoning-enabled group shows smaller internal gaps, especially among the three latest models. The highest score is 88.99 (Kimi-K2.5), with Qwen3.5-Plus also very strong at 88.10. This shows that the reasoning-enabled models have more stable performance and higher overall levels.

As shown in Table 3, the three top-performing reasoning-enabled models, including Qwen3.5-Plus, DeepSeek-V3.2, and Kimi-K2.5, exhibit relatively small performance gaps. Therefore, it is important to determine whether such differences are statistically significant rather than attributable to random variation. To this end, McNemar's test [59], a non-parametric statistical method for paired nominal data, is applied to perform pairwise comparisons between the models on the same mathematical problems. The continuity-corrected McNemar statistic is defined as

$$\chi^2 = \frac{(|b - c| - 1)^2}{(b + c)} \tag{6}$$

where b denotes the number of problems correctly solved only by Model A and c denotes the number of problems correctly solved only by Model B. The p-value is computed from the chi-square distribution with one degree of freedom. The conventional significance threshold of $p < 0.05$ is adopted in this study.

The comparison is conducted using paired question-level binary correctness outcomes over all 3429 mathematical problems rather than macro-average course rankings. The results are summarized in Table 5.

Table 5. McNemar's test results among top-performing reasoning-enabled LLMs.

Model A	Model B	A Correct/B Wrong	A Wrong/B Correct	McNemar Statistic	p-Value
Qwen3.5-Plus	DeepSeek-V3.2	65	55	0.6750	0.4113
Qwen3.5-Plus	Kimi-K2.5	50	70	3.0083	0.0828
DeepSeek-V3.2	Kimi-K2.5	45	75	7.0083	0.0081

As shown in Table 5, Qwen3.5-Plus and DeepSeek-V3.2 do not show a statistically significant difference ($p = 0.4113$), indicating highly comparable problem-solving performance. In contrast, Kimi-K2.5 demonstrates a statistically significant advantage over DeepSeek-V3.2 ($p = 0.0081$). The comparison between Kimi-K2.5 and Qwen3.5-Plus shows a marginally significant trend ($p = 0.0828$), suggesting that Kimi-K2.5 consistently achieves slightly stronger problem-solving performance across the evaluated datasets, although the difference does not reach the conventional statistical significance threshold ($p < 0.05$).

4.2.3. Course-Level Comparison

To further analyze each LLM's performance across the five core curricula, course-level accuracy scores are calculated using Formula (1) and summarized in Table 6.

Table 6. Accuracy scores of all LLMs across five core mathematics curricula [1].

Math Curriculum	Claude-3.5-Haiku	GPT-4o-Mini	DeepSeek-V3	GPT-4.1	O3-Mini	Qwen-Plus	DeepSeek-R1	O4-Mini	Qwen3.5-Plus	DeepSeek-V3.2	Kimi-K2.5
Algebra	71.23	79.04	90.16	92.3	93.16	91.76	91.87	93.9	94.33	94.12	**94.65**
Calculus	37.82	47.55	76.42	82.83	86.39	86.87	87.34	88.45	91.06	90.11	**91.46**
Differential Equations	21.98	31.27	60.06	70.28	77.71	76.78	78.02	77.4	78.95	80.5	**81.11**
Precalculus	57.95	69.45	83.42	86.3	88.77	88.9	89.18	88.49	89.73	89.45	**90.14**
Probability and Statistics	49.72	56.5	72.88	83.05	81.92	83.62	84.18	85.31	86.44	87.01	**87.57**
Total	47.74	56.76	76.59	82.95	85.59	85.59	86.12	86.71	88.1	88.24	**88.99**

[1] The highest values are marked in bold and the second-highest ones are underlined.

To better show course-level scores, Figure 3 also presents a line chart of score trends. The chart clearly reflects performance differences across curricula and the relative difficulty of each course for LLMs.

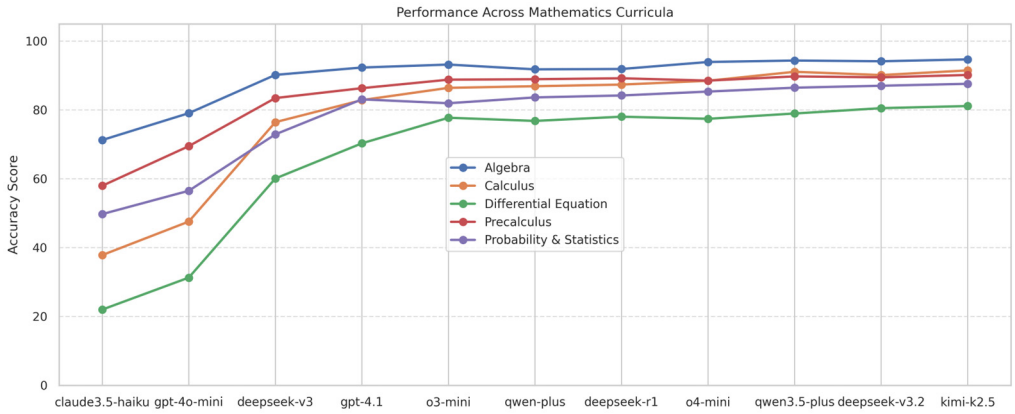


Figure 3. Course-level accuracy score of all LLMs.

As shown in Table 6 and Figure 3, the difficulty of the five core courses varies significantly, and all the models show a consistent performance trend across courses: the average score is highest in Algebra and lowest in Differential Equations, reflecting differing difficulty levels. Specifically, Kimi-K2.5 achieves 94.65 in Algebra, significantly higher than its other course scores. This indicates that university-level Algebra problems are relatively easy for LLMs. In contrast, the highest score in Differential Equations is only 81.11, the lowest among the five courses. This shows that Differential Equations demands stronger logical reasoning and professional knowledge, making it the most challenging course for all the tested LLMs. A detailed case study of Differential Equations appears in Section 4.2.6.

In the non-reasoning group, GPT-4.1 performs well in each course under the evaluated setting: 92.3 (Algebra), 82.83 (Calculus), 70.28 (Differential Equations), 86.3 (Precalculus), and 83.05 (Probability and Statistics). However, the non-reasoning models are mostly outperformed by the reasoning-enabled models. This confirms that LLM reasoning ability is a key factor affecting performance on university-level mathematics problems.

In the reasoning-enabled group, model performance varies slightly across courses but remains high overall. Kimi-K2.5 ranks first in both overall and individual course performance. Qwen3.5-Plus and DeepSeek-V3.2 also perform well, second only to Kimi-K2.5. These results indicate that the reasoning-enabled models exhibit different strengths in specific curricula, and their reasoning capabilities have steadily improved from 2024 to 2026, highlighting the rapid pace of recent advancements.

4.2.4. Evaluator Analysis

To evaluate the reliability of the seven reasoning-enabled evaluator models, two metrics defined with Formulas (4) and (5) are adopted. Unlike curriculum-specific analysis, this section integrates all the questions, models, and curricula into a unified dataset to quantify consistency between each evaluator's grading and the final ensemble score. This global statistical approach avoids bias from unbalanced course sample sizes and more accurately reflects intrinsic evaluator reliability.

The detailed agreement rate and Kappa coefficient results are summarized in Table 7.

To better visualize reliability differences among the seven evaluators, Figure 4 presents a heatmap of agreement rate and Kappa coefficient, with colors representing metric values sorted in descending order. The visualization intuitively shows consistency gaps between each evaluator and the ensemble score.

Table 7. Agreement rate and Kappa coefficient of evaluator models [1].

Evaluator Model	Agreement Rate	95% CI	Kappa Coefficient	95% CI
O3-Mini	0.9692	[0.9676, 0.9710]	0.8973	[0.8916, 0.9032]
Qwen-Plus	0.9724	[0.9708, 0.9740]	0.9111	[0.9059, 0.9165]
DeepSeek-R1	0.9621	[0.9602, 0.9641]	0.8775	[0.8713, 0.8838]
O4-Mini	**0.9788**	[0.9773, 0.9801]	**0.9303**	[0.9254, 0.9346]
Qwen3.5-Plus	0.9632	[0.9613, 0.9652]	0.8681	[0.8610, 0.8750]
DeepSeek-V3.2	<u>0.9733</u>	[0.9717, 0.9750]	<u>0.9146</u>	[0.9093, 0.9201]
Kimi-K2.5	0.9722	[0.9706, 0.9739]	0.9028	[0.8970, 0.9088]

[1] The highest values are marked in bold and the second-highest ones are underlined.

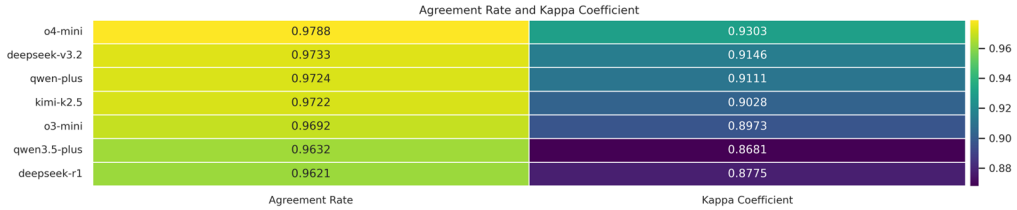


Figure 4. Agreement rate and Kappa coefficient.

As shown in Table 7 and Figure 4, all seven evaluators show high consistency with the ensemble score in the global merged calculation, demonstrating good reliability. Specifically, the agreement rate ranges from 0.9621 (DeepSeek-R1) to 0.9788 (O4-Mini), all above 0.96. The Kappa coefficient further verifies reliability, ranging from 0.8681 (Qwen3.5-Plus) to 0.9303 (O4-Mini), all above 0.85. According to general academic standards [60], Kappa values between 0.8 and 1.0 indicate near-perfect consistency, proving that the seven evaluators not only align highly with the ensemble score but also maintain stable, unified grading standards in the global dataset.

To further assess statistical robustness, 95% CIs are computed using bootstrap resampling. As shown in Table 7, all the evaluators exhibit extremely narrow confidence intervals for both agreement rate and Kappa coefficient, suggesting that evaluator consistency remains highly stable across different bootstrap samples. For example, O4-Mini achieves an agreement rate of 0.9788 with a 95% CI of [0.9773, 0.9801], while DeepSeek-V3.2 achieves 0.9733 with a 95% CI of [0.9717, 0.9750].

Although O4-Mini achieves the highest overall agreement rate and Kappa coefficient, the confidence intervals of several top-performing evaluators partially overlap, indicating that their evaluator reliability is broadly comparable under the current benchmark setting. Overall, O4-Mini, DeepSeek-V3.2, and Qwen-Plus demonstrate the strongest alignment with the ensemble score and maintain consistently high evaluator stability.

Notably, the global merged evaluation further supports the rationality of the multi-evaluator ensemble scheme adopted in this study. Even the most reliable single evaluator (O4-Mini) still shows a measurable agreement rate gap compared with the seven-evaluator ensemble, suggesting that no individual evaluator can fully replace the robustness provided by ensemble-based assessment.

Taken together, the evaluator analysis confirms that reasoning-enabled LLMs can provide highly stable and reliable automatic assessment for university-level mathematical problem solving while also supporting the necessity of the proposed multi-evaluator ensemble framework.

4.2.5. Cost Comparison

Beyond reasoning ability and evaluator reliability, API service cost is also an important consideration in practical applications. This section compares LLM costs across two key stages: the answering stage and the evaluation stage. In the first stage, eleven LLMs generate responses for all the math problems; in the second stage, the seven selected evaluators judge the correctness of all the generated answers. Cost indicators include input price, output price, total token consumption, and total cost, all based on actual API service data for each model. Pricing information was collected in April 2026 from publicly accessible APIs and may vary across third-party platforms and time periods. Inference cost is influenced by deployment strategies, API pricing policies, and token efficiency associated with different service providers.

Table 8 shows specific cost data for all the LLMs in the answering stage, including input price (USD per 1 million tokens, with CNY equivalents), output price (USD per 1 million tokens), total tokens (millions), and total cost (USD).

Table 8. Cost comparison of LLMs in the answering stage.

Model	Input Price (USD $/M)	Output Price (USD $/M)	Total Tokens [1] (M)	Total Cost (USD $)
Claude-3.5-Haiku	0.25	1.25	2.50	9.86
GPT-4o-Mini	0.15	0.6	3.71	1.85
DeepSeek-V3	0.28 (¥2 CNY)	1.14 (¥8 CNY)	3.67	3.46 (¥24.52 CNY)
GPT-4.1	2	8	4.51	31.11
O3-Mini	1.1	4.4	6.59	26.28
Qwen-Plus	0.11 (¥0.8 CNY)	2.28 (¥16 CNY)	16.96	36.95 (¥258.69 CNY)
DeepSeek-R1	0.57 (¥4 CNY)	2.28 (¥16 CNY)	19.12	42.26 (¥295.82 CNY)
O4-Mini	1.1	4.4	5.57	21.83
Qwen3.5-Plus	0.11 (¥0.8 CNY)	0.68 (¥4.8 CNY)	18.81	12.43 (¥87.01 CNY)
DeepSeek-V3.2	0.28 (¥2 CNY)	0.42 (¥3 CNY)	12.66	5.31 (¥37.17 CNY)
Kimi-K2.5	0.57 (¥4 CNY)	3 (¥21 CNY)	13.78	39.32 (¥275.29 CNY)

[1] Token counts are reported in "M," where "M" denotes million (1 M = 1,000,000 tokens).

As shown in Table 8, total cost varies significantly across LLMs in the answering stage. Non-reasoning group models have relatively low cost except for GPT-4.1, mainly because they generate few reasoning traces, resulting in low output token consumption. In contrast, most reasoning-enabled models cost over $12, except DeepSeek-V3.2. DeepSeek-R1 has the highest total cost at $42.26, followed by Kimi-K2.5 ($39.32) and Qwen-Plus ($36.95). The high cost of these models stems mainly from large output token consumption and relatively high output prices.

Notably, DeepSeek-V3.2, a high-performance reasoning model, has the lowest cost among top reasoning models at $5.31. Referring back to overall accuracy scores in Table 3, DeepSeek-V3.2 scores 88.24, just 0.79 points below Kimi-K2.5's top score of 88.99. However, Kimi-K2.5 costs $39.32, seven times that of DeepSeek-V3.2. In the evaluation stage, specific cost data for the seven evaluator models are summarized in Table 9.

Table 9. Cost comparison of LLMs in the evaluation stage.

Model	Input Price (USD $/M)	Output Price (USD $/M)	Total Tokens (M)	Total Cost (USD $)
O3-Mini	1.1	4.4	6.41	20.32
Qwen-Plus	0.11 (¥0.8 CNY)	2.24 (¥16 CNY)	11.53	20.17 (¥143.59 CNY)
DeepSeek-R1	0.56 (¥4 CNY)	2.24 (¥16 CNY)	15.84	31.70 (¥225.69 CNY)
O4-Mini	1.1	4.4	4.93	13.69
Qwen3.5-Plus	0.11 (¥0.8 CNY)	0.68 (¥4.8 CNY)	26.44	18.28 (¥127.93 CNY)
DeepSeek-V3.2	0.28 (¥2 CNY)	0.42 (¥3 CNY)	11.59	4.58 (¥32.06 CNY)
Kimi-K2.5	0.57 (¥4 CNY)	3 (¥21 CNY)	14.33	36.27 (¥253.88 CNY)

According to Table 9, cost differences among the seven evaluators in the evaluation stage are also significant. DeepSeek-V3.2 has the lowest total cost ($4.58), mainly due to low input/output prices and efficient token usage.

O3-Mini and O4-Mini have moderate total costs ($20.32 and $13.69 respectively). Kimi-K2.5 and DeepSeek-R1 have higher evaluation-stage costs ($36.27 and $31.70 respectively); their high costs limit application in large-scale automated evaluation scenarios. Qwen-Plus and Qwen3.5-Plus have moderate costs among the seven evaluators ($20.17 and $18.28 respectively).

From a technical perspective, DeepSeek-V3.2's optimized attention mechanism allows it to quickly focus on key information during evaluation, avoiding redundant token processing. Its lightweight structure also reduces computational overhead, reflected in lower API pricing and more efficient token consumption, making it the most cost-effective model in the evaluation stage.

Combining two-stage cost analysis, cost is clearly an important constraint in practical LLM applications. Under the evaluated conditions, DeepSeek-V3.2 demonstrates a competitive balance between performance and inference cost and achieves the most reasonable accuracy–cost trade-off. Its cost-performance may stem from underlying technical optimizations shown in its technical report [36]; it uses a streamlined architecture based on efficient transformer variants, reducing redundant parameter computation while maintaining high reasoning accuracy. Additionally, its token encoding mechanism is optimized for mathematical reasoning, enabling more concise and efficient output of reasoning processes. These technical advantages allow DeepSeek-V3.2 to achieve a better balance between performance and cost, making it valuable for undergraduate math problem-solving tasks.

4.2.6. Case Study of Course

This section focuses on special phenomena identified in the course-level results, especially the poor performance of all the LLMs on Differential Equations. An in-depth case study of over 100 mismatches between the model outputs and golden answers analyzes potential causes, integrating inherent characteristics of Differential Equations and intrinsic limitations of LLMs.

One of the most critical and frequent error types is solution structure mismatch, where models fail to correctly identify the expected solution form or nature. Some models incorrectly output general homogeneous solutions when the task requires a particular non-homogeneous solution. In other cases, models produce particular solutions instead of the required general solutions.

For example, solving Question_3871_exercise.5.1.11 in CollegeMath, LLMs generate $y_2 = e^{6x}$, while the golden answer is $y_2 = xe^{3x}$. This reveals that LLMs may fail to construct linearly independent solutions under repeated roots. Unlike other math courses, Differential Equations inherently require precise identification of solution spaces and appropriate functional forms. As a result, solution structure mismatch becomes a typical error type.

Another distinguishing feature of Differential Equations is high representational flexibility. A single solution can often be expressed in multiple mathematically equivalent forms, such as exponential vs. logarithmic, trigonometric linear combination vs. amplitude-phase, and implicit vs. explicit.

For example, solving Question_3795_exercise.4.5.28 in CollegeMath, the required golden answer is $e^{2y^2} = k(1 + 2x^2)$, while some LLMs produce $2y^2 - \ln(1 + 2x^2) = C$. In fact, these two forms represent the same solution. The LLM outputs may appear quite different but can still be mathematically correct. Although mathematical equivalence was emphasized in evaluation prompt design, LLMs frequently struggle to maintain equivalence consistency, especially for flexible solution forms in Differential Equations. This kind of equivalence ambiguity further complicates automatic evaluation and undermines the stability of model scoring.

Additionally, other typical errors occur in sensitive eigenvalues or frequencies during intermediate calculation steps. LLMs often generate incorrect exponential rates or wrong frequencies; even small parameter mistakes can lead to entirely different solutions, further reducing accuracy on Differential Equations.

The relatively lower performance observed in Differential Equations may be influenced not only by the intrinsic complexity of the subject, but also by the limited diversity and potential sampling bias in the dataset composition. All 323 Differential Equations problems are drawn from the CollegeMath dataset, which may introduce dataset-specific bias and reduce the generalizability of the findings. As a result, the conclusion that Differential Equations is the most challenging course for current LLMs should be interpreted with caution. Future work should incorporate more diverse problem sources and broader course-level evaluations to further validate this observation.

4.2.7. Ablation Study

Human Validation on Subset100

To further analyze the reliability of the proposed ensemble-based evaluation framework, an additional human validation experiment was conducted on a manually selected subset of 100 representative problems, denoted as Subset100. The subset was constructed by uniformly sampling 20 problems from each of the five undergraduate mathematics curricula, including Algebra, Calculus, Differential Equations, Precalculus, and Probability and Statistics. The primary purpose of this experiment is to examine whether the automated ensemble evaluation can achieve grading consistency comparable to expert human assessment.

For each problem in Subset100, the responses generated by all eleven LLMs were independently graded by two university mathematics professors under a double-blind evaluation protocol with reference to the golden answers. The human experts were unaware of the model identities during scoring, and disagreements were further cross-checked to ensure grading consistency. The corresponding human evaluation scores are compared with the ensemble-based evaluation scores for each of the five curricula in Subset100, presented with two heatmaps in Figures 5 and 6, with colors representing score values sorted in ascending order.

As shown in Figures 5 and 6, the course-level score distributions produced by the ensemble framework closely follow those obtained from human expert grading across all the curricula and models. The ranking trends among the LLMs remain highly consistent between the two evaluation schemes. For example, the reasoning-enabled models consistently outperform the non-reasoning models under both ensemble and human evaluation, while Differential Equations remains the most challenging curriculum for most tested LLMs. Only minor score deviations can be observed in several difficult subjects, particularly Differential

Equations, where equivalent symbolic forms and diverse derivation styles may increase grading ambiguity.

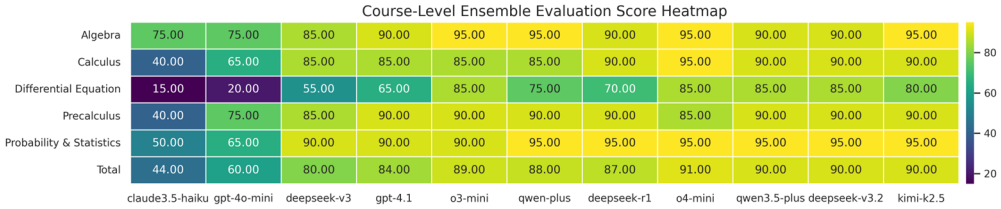


Figure 5. Course-level ensemble evaluation score heatmap.

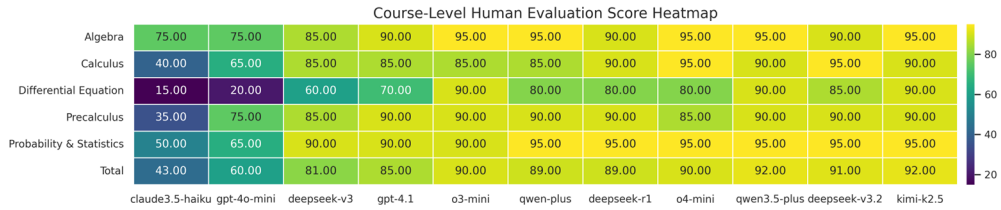


Figure 6. Course-level human evaluation score heatmap.

To quantitatively assess agreement between the ensemble evaluation and human grading, agreement rate and Cohen's Kappa coefficient were computed across all the subjects. The detailed results are summarized in Table 10.

Table 10. Agreement rate and Kappa coefficient in Subset100.

Math Curriculum	Agreement Rate	Kappa Coefficient
Algebra	0.9955	0.9675
Calculus	0.9955	0.9675
Differential Equations	0.9455	0.8159
Precalculus	0.9955	0.9903
Probability and Statistics	1.0000	1.0000
Total	0.9864	0.9483

The results show high consistency between the ensemble evaluation framework and expert human grading. Specifically, the overall agreement rate reaches 0.9864, while the overall Kappa coefficient reaches 0.9483, indicating near-perfect agreement according to standard interpretation criteria [60]. Across four curricula, the agreement rate exceeds 99.5%, and Probability and Statistics achieves complete agreement between ensemble evaluation and human experts.

Although Differential Equations shows relatively lower consistency compared with the other curricula, its agreement rate and Kappa coefficient still reach 0.9455 and 0.8159 respectively, remaining within the range of strong agreement. This phenomenon is likely related to the greater symbolic variability and equivalent-form transformations commonly encountered in Differential Equations problems described in Section 4.2.6, which may introduce additional ambiguity in automated answer matching.

Furthermore, the overall score distributions produced by the ensemble framework exhibit trends highly consistent with those obtained from human grading, indicating that the proposed ensemble strategy can reliably approximate expert-level evaluation behavior. These findings support the effectiveness and practical reliability of the proposed multi-evaluator ensemble framework for large-scale automated assessment of university-level mathematical problem solving.

Ablation on Evaluator Ensemble Size

To further investigate the effect of evaluator ensemble size on grading stability, an ablation experiment is conducted by gradually reducing the number of evaluators participating in the ensemble process. Considering the advantage of odd-sized evaluator ensembles in avoiding tie votes, this experiment focuses on evaluator combinations of size 1, 3, 5, and 7. The primary objective is to analyze how evaluator count influences consistency and robustness in automatic mathematical assessment.

Here, the ensemble result from the full seven evaluators is treated as ground truth. Agreement rate and Kappa coefficient are also used as core metrics to quantify reliability gaps across evaluator combinations. The detailed ablation results are summarized in Table 11.

Table 11. Results of ablation experiment on evaluator count.

Number of Evaluators	Mean Agreement Rate	Std [1] of Agreement Rate	Mean Kappa Coefficient	Std [1] of Kappa Coefficient
1	0.9702	0.0055	0.9002	0.0200
3	0.9847	0.0036	0.9489	0.0126
5	0.9921	0.0018	0.9737	0.0061
7	1.0000	0.0000	1.0000	0.0000

[1] Std means standard deviation.

As the number of evaluators increases, both the agreement rate and Kappa coefficient consistently improve, while the corresponding standard deviations decrease. Specifically, using only one evaluator shows a relatively high mean agreement rate (0.9702) and Kappa coefficient (0.9002), but standard deviations (0.0055 and 0.0200) reveal inherent individual bias and instability in single-evaluator judgments. This suggests that relying on a single evaluator may introduce higher variance and evaluator-specific bias, underscoring the need for an ensemble scheme to mitigate such biases.

When the number of evaluators increases to 3, consistency improves with reduced standard deviations. This improvement demonstrates the advantages of the ensemble scheme. However, even the three-evaluator ensemble still lags behind the full ensemble. A similar trend is observed for the five-evaluator ensemble.

In summary, the ablation results show that increasing the number of evaluators generally improves grading consistency and reduces evaluator-specific variance. Although individual evaluators and smaller evaluator combinations already show relatively high reliability, they remain more sensitive to evaluator-specific variations than larger ensembles. These results suggest that larger evaluator ensembles generally provide more stable grading behavior.

4.2.8. Study Limitations

Despite the encouraging results, this study still has several limitations. First, the comparative analysis mainly focuses on text-based mathematical problems and does not include geometric figures or multimodal mathematical reasoning tasks. In addition, potential overlap between public datasets and model pretraining corpora cannot be completely excluded. Such contamination may partially influence the observed reasoning and problem-solving performance differences among LLMs.

Second, the evaluation framework still relies heavily on LLM-based judgment, which may introduce systematic evaluator bias despite the use of ensemble voting strategies. Although limited human validation was conducted on a representative subset of problems, broader expert-level verification remains necessary for further reliability assessment.

Third, the distribution of problems across curricula is uneven, and the diversity of some subjects remains limited. In particular, Differential Equations and Probability and Statistics are underrepresented compared with Calculus and Algebra, which may affect aggregate performance comparisons and the generalizability of certain course-level conclusions. Future studies could incorporate additional public datasets with more balanced curriculum coverage, especially for Differential Equations and Probability and Statistics.

In addition, the current study adopts a unified prompting strategy across all the evaluated LLMs in order to maintain consistent experimental conditions. Although this design improves fairness and comparability, different models may exhibit varying sensitivity to prompt structure and instructional style. The reported results should be interpreted as valid under the evaluated prompting setting. Future work may further investigate the influence of prompt design on university-level mathematical tasks.

Finally, the current evaluation primarily focuses on final-answer correctness and does not fully assess the mathematical rigor or validity of the intermediate reasoning steps. Future work may incorporate step-level reasoning evaluation and formal proof verification to provide a more comprehensive assessment of university-level mathematical reasoning.

5. Conclusions and Answers to Research Questions

This study provides a curriculum-aware evaluation of modern LLMs for university-level mathematics, supported by an automated inference and evaluation pipeline with an ensemble-based assessment framework. Based on extensive experiments across three public datasets and five core undergraduate courses, the answers to the three research questions are summarized below.

(RQ1) How do state-of-the-art LLMs perform across core undergraduate mathematics courses?

State-of-the-art LLMs exhibit clear performance variations across different undergraduate mathematics courses. All the models achieve the highest accuracy in Algebra and Precalculus, moderate performance in Calculus and Probability and Statistics, and the lowest accuracy in Differential Equations. Such patterns indicate that solution structure complexity, symbolic reasoning requirements, and representational flexibility in advanced courses pose significant challenges to current LLMs.

(RQ2) What performance differences exist between reasoning-enabled LLMs and conventional non-reasoning LLMs on university-level mathematical problem-solving?

Reasoning-enabled LLMs generally achieve higher overall performance than non-reasoning models under the evaluated experimental setting. Top reasoning models such as Kimi-K2.5 and DeepSeek-V3.2 achieve the highest scores, while non-reasoning models exhibit comparatively weaker robustness in multi-step and symbolic mathematical tasks. The performance gap suggests that reasoning-oriented optimization strategies are associated with stronger mathematical problem-solving performance in the evaluated tasks.

(RQ3) How effective is the ensemble evaluation paradigm using multiple LLMs as evaluators?

The proposed ensemble evaluation paradigm shows high agreement with expert grading and provides relatively stable evaluation results. All seven reasoning-enabled evaluators achieve high agreement rates and Kappa coefficients, and the ensemble scheme effectively reduces individual bias and judgment instability. Ablation studies further confirm that a full set of evaluators yields the most stable and consistent evaluation behavior.

In summary, this study quantitatively analyzes the strengths and limitations of modern LLMs in undergraduate mathematics and validates the effectiveness of the proposed ensemble-based evaluation framework. Future work will extend evaluation to multimodal

mathematical problems involving diagrams, charts, and geometric figures, and explore more efficient models for educational deployment.

Author Contributions: Conceptualization, F.Y.; methodology, F.Y.; software, F.Y. and W.Q.; validation, F.Y., W.X. and W.Q.; writing—original draft preparation, F.Y. and W.Q.; writing—review and editing, F.Y., W.X. and W.Q.; visualization, F.Y.; supervision, F.Y. and W.X.; project administration, F.Y. and W.X.; funding acquisition, F.Y. and W.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the JIANGSU PROVINCE HIGHER EDUCATION "HIGH-QUALITY PUBLIC COURSE TEACHING REFORM RESEARCH" SPECIAL PROJECT with grant number 2025GZJX038 and the NANJING TECH UNIVERSITY KEY GENERAL PUBLIC BASIC COURSE REFORM PROJECT with grant number 20250009.

Data Availability Statement: This study utilizes the three public university-level mathematical datasets, which are made available under a specific license. Researchers interested in accessing these datasets could refer to their official download website. Additionally, the example codes and inference solution results in this paper will be publicly available at the following GitHub repository: https://github.com/fftechnj/umath-problem-llm-evaluation (accessed on 13 May 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AIME	American Invitational Mathematics Examination
CI	Confidence Interval
CoT	Chain-of-Thought
DSA	DeepSeek Sparse Attention
LLMs	Large Language Models
MoE	Mixture-of-Experts
NLP	Natural Language Processing
RLHF	Reinforcement Learning from Human Feedback

References

1. Karanikolas, N.; Manga, E.; Samaridi, N.; Tousidou, E.; Vassilakopoulos, M. Large language models versus natural language understanding and generation. In Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics (PCI 2023), Lamia, Greece, 24–26 November 2023.
2. Min, B.; Ross, H.; Sulem, E.; Veyseh, A.P.B.; Nguyen, T.H.; Sainz, O.; Agirre, E.; Heintz, I.; Roth, D. Recent advances in natural language processing via large pretrained language models: A survey. *ACM Comput. Surv.* **2023**, *56*, 30. [CrossRef]
3. Liu, J.; Xia, C.S.; Wang, Y.; Zhang, L. Is your code generated by ChatGPT really correct? Rigorous evaluation of large language models for code generation. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 21558–21572.
4. Jiang, J.; Wang, F.; Shen, J.; Kim, S.; Kim, S. A survey on large language models for code generation. *ACM Trans. Softw. Eng. Methodol.* **2024**, *35*, 58. [CrossRef]
5. Zhang, Y.; Chen, X.; Jin, B.; Wang, S.; Ji, S.; Wang, W.; Han, J. A comprehensive survey of scientific large language models and their applications in scientific discovery. In Proceedings of the EMNLP 2024, Miami, FL, USA, 12–16 November 2024; pp. 8783–8817.
6. Li, S.; Puig, X.; Paxton, C.; Du, Y.; Wang, C.; Fan, L.; Chen, T.; Huang, D.A.; Akyürek, E.; Anandkumar, A.; et al. Pre-trained language models for interactive decision making. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 31199–31212.
7. Song, Z.; Xu, N.; Ding, H. Decision-making framework for autonomous vehicles in complex scenarios using large language models. *Automot. Innov.* **2026**, *9*, 1–21. [CrossRef]
8. Kalyan, K.S. A survey of GPT-3 family large language models including ChatGPT and GPT-4. *Nat. Lang. Process. J.* **2024**, *6*, 100048. [CrossRef]
9. Qwen Team. Qwen2.5 technical report. *arXiv* **2024**, arXiv:2412.15115.
10. DeepSeek-AI; Liu, B.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Dengr, C.; Ruan, C.; Dai, D.; et al. DeepSeek-V2: A strong, economical, and efficient mixture-of-experts language model. *arXiv* **2024**, arXiv:2405.04434.

11. Guo, D.; Yang, D.; Zhang, H.; Song, J.; Wang, P.; Zhu, Q.; Xu, R.; Zhang, R.; Ma, S.; Bi, X.; et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv* **2025**, arXiv:2501.12948.
12. Chu, Z.; Wang, S.; Xie, J.; Zhu, T.; Yan, Y.; Ye, J.; Zhong, A.; Hu, X.; Liang, J.; Yu, P.S.; et al. LLM agents for education: Advances and applications. *arXiv* **2025**, arXiv:2503.11733.
13. Al Faraby, S.; Romadhony, A.; Adiwijaya. Analysis of LLMs for educational question classification and generation. *Comput. Educ. Artif. Intell.* **2024**, *7*, 100298. [CrossRef]
14. Khan, A.A.; Tabassum, M.A.; Umar, Z. Effect of Magic School AI tool on elementary students' academic achievement. *Al Khadim Res. J. Islam. Cult. Civ.* **2024**, *5*, 13–28.
15. Shochcho, M.I.; Rahman, M.A.U.; Rohan, S.; Islam, A.; Heickal, H.; Rahman, A.K.M.M.; Amin, M.A.; Ali, A.A. Improving user engagement and learning outcomes in LLM-based Python tutor: A study of PACE. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 26 April–1 May 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 1–12.
16. Hu, B.; Zhu, J.; Pei, Y.; Gu, X. Exploring the potential of LLM to enhance teaching plans through teaching simulation. *npj Sci. Learn.* **2025**, *10*, 7. [CrossRef] [PubMed]
17. Shi, Y.; Yu, K.; Dong, Y.; Chen, F. Large language models in education: A systematic review of empirical applications, benefits, and challenges. *Comput. Educ. Artif. Intell.* **2026**, *10*, 100529. [CrossRef]
18. Imamguluyev, R.; Hasanova, P.; Imanova, T.; Mammadova, A.; Hajizada, S.; Samadova, Z. AI-powered educational tools: Transforming learning in the digital era. *Int. Res. J. Mod. Eng. Technol. Sci.* **2024**, *6*, 920–929.
19. Liu, V.; Latif, E.; Zhai, X. Advancing education through tutoring systems: A systematic literature review. *arXiv* **2025**, arXiv:2503.09748. [CrossRef]
20. Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. Training verifiers to solve math word problems. *arXiv* **2021**, arXiv:2110.14168. [CrossRef]
21. Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. *arXiv* **2021**, arXiv:2103.03874. [CrossRef]
22. He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level multimodal problems. *arXiv* **2024**, arXiv:2402.14008.
23. Huang, K.; Guo, J.; Li, Z.; Ji, X.; Ge, J.; Li, W.; Guo, Y.; Cai, T.; Yuan, H.; Wang, R.; et al. Math-Perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. *arXiv* **2025**, arXiv:2502.06453. [CrossRef]
24. Fang, M.; Wan, X.; Lu, F.; Xing, F.; Zou, K. MathOdyssey: Benchmarking mathematical problem-solving skills in large language models. *arXiv* **2024**, arXiv:2406.18321.
25. Chernyshev, K.; Polshkov, V.; Stepanov, V.; Myasnikov, A.; Artemova, E.; Miasnikov, A.; Tilga, S. U-MATH: A university-level benchmark for evaluating mathematical skills in large language models. In Proceedings of the GEM2, Vienna, Austria, 31 July–1 August 2025; pp. 974–1001.
26. Tang, Z.; Zhang, X.; Wang, B.; Wei, F. MathScale: Scaling instruction tuning for mathematical reasoning. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; PMLR 235:47885-47900.
27. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 5998–6008.
28. Koubaa, A. GPT-4 vs. GPT-3.5: A concise showdown. *Preprints* **2023**, 2023030422. [CrossRef]
29. OpenAI. OpenAI o3-Mini: Pushing the Frontier of Cost-Effective Reasoning. Available online: https://openai.com/index/openai-o3-mini/ (accessed on 16 December 2025).
30. OpenAI. Introducing OpenAI o3 and o4-Mini. Available online: https://openai.com/index/introducing-o3-and-o4-mini/ (accessed on 16 December 2025).
31. Anthropic. Model Card Addendum: Claude 3.5 Haiku and Upgraded Claude 3.5 Sonnet. Available online: https://assets.anthropic.com (accessed on 16 December 2025).
32. Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Lu, K.; et al. Qwen2.5-Coder technical report. *arXiv* **2024**, arXiv:2409.12186. [CrossRef]
33. Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; et al. Qwen technical report. *arXiv* **2023**, arXiv:2309.16609. [CrossRef]
34. Team, K.; Bai, T.; Bai, Y.; Bao, Y.; Cai, S.H.; Cao, Y.; Charles, Y.; Che, H.S.; Chen, C.; Chen, G.; et al. Kimi K2.5: Visual agentic intelligence. *arXiv* **2026**, arXiv:2602.02276. [CrossRef]
35. Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C. DeepSeek-V3 technical report. *arXiv* **2024**, arXiv:2412.19437.
36. Liu, A.; Mei, A.; Lin, B.; Xue, B.; Wang, B.; Xu, B.; Wu, B. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv* **2025**, arXiv:2512.02556.
37. Ni, S.; Chen, G.; Li, S.; Chen, X.; Li, S.; Wang, B.; Wang, Q.; Wang, X.; Zhang, Y.; Fan, L.; et al. A survey on large language model benchmarks. *arXiv* **2025**, arXiv:2508.15361. [CrossRef]

38. Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E.H.; Schärli, N.; Zhou, D. Large language models can be easily distracted by irrelevant context. *arXiv* **2023**, arXiv:2302.00093. [CrossRef]
39. Mirzadeh, I.; Alizadeh, K.; Shahrokhi, H.; Tuzel, O.; Bengio, S.; Farajtabar, M. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv* **2024**, arXiv:2410.05229.
40. Srivastava, S.; Annarose, M.B.; Anto, P.V.; Menon, S.; Sukumar, A.; Adwaith Samod, T.; Philipose, A.; Prince, S.; Thomas, S. Functional benchmarks for robust evaluation of reasoning performance. *arXiv* **2024**, arXiv:2402.19450. [CrossRef]
41. Lewkowycz, A.; Andreassen, A.J.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. Solving quantitative reasoning problems with language models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 3843–3857.
42. Liu, O.L.; Rios, J.A.; Heilman, M.; Gerard, L.; Linn, M.C. Validation of automated scoring of science assessments. *J. Res. Sci. Teach.* **2016**, *53*, 215–233. [CrossRef]
43. Stephan, A.; Zhu, D.; Aßenmacher, M.; Shen, X.; Roth, B. From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks. *arXiv* **2024**, arXiv:2409.04168.
44. Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; Liu, Y. LLMs-as-judges: A comprehensive survey on LLM-based evaluation methods. *arXiv* **2024**, arXiv:2412.05579.
45. Li, D.; Jiang, B.; Huang, L.; Beigi, A.; Zhao, C.; Tan, Z.; Bhattacharjee, A.; Jiang, Y.; Chen, C.; Wu, T.; et al. From generation to judgment: Opportunities and challenges of LLM-as-a-judge. In Proceedings of the EMNLP, Suzhou, China, 4–9 November 2025; pp. 2757–2791.
46. Dikli, S. An overview of automated scoring of essays. *J. Technol. Learn. Assess.* **2006**, *5*.
47. Baral, S.; Botelho, A.F.; Erickson, J.A.; Benachamardi, P.; Heffernan, N.T. Improving automated scoring of student open responses in mathematics. *Int. Educ. Data Min. Soc.* **2021**.
48. Gupta, A.; Reddig, J.; Calò, T.; Weitekamp, D.; MacLellan, C.J. Beyond final answers: Evaluating large language models for math tutoring. In *Artificial Intelligence in Education*; Springer: Cham, Switzerland, 2025; pp. 323–337.
49. Baral, S.; Worden, E.; Lim, W.C.; Luo, Z.; Santorelli, C.; Ashish, G. Automated assessment in math education: A comparative analysis of LLMs for open-ended responses. In Proceedings of the International Conference on Educational Data Mining, Atlanta, GA, USA, 14–17 July 2024.
50. Lee, U.; Kim, Y.; Lee, S.; Park, J.; Mun, J.; Lee, E.; Kim, H.; Lim, C.; Yoo, Y.J. Can we use GPT-4 as a mathematics evaluator in education? *Int. J. Artif. Intell. Educ.* **2024**, *35*, 1560–1596. [CrossRef]
51. Xia, S.; Li, X.; Liu, Y.; Wu, T.; Liu, P. Evaluating mathematical reasoning beyond accuracy. *arXiv* **2024**, arXiv:2404.05692. [CrossRef]
52. Liu, T.; Chen, Z.; Fang, Z.; Luo, W.; Tian, M.; Liu, Z. MathEval: A comprehensive benchmark for evaluating large language models. *Front. Digit. Educ.* **2025**, *2*, 16. [CrossRef]
53. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 46595–46623.
54. Espejel, J.L.; Ettifouri, E.H.; Alassan, M.S.Y.; Chouham, E.M.; Dahhane, W. GPT-3.5, GPT-4, or BARD? Evaluating LLM reasoning ability. *Nat. Lang. Process. J.* **2023**, *5*, 100032. [CrossRef]
55. Zhang, C.; Liu, H.; Zeng, J.; Yang, K.; Li, Y.; Li, H. Prompt-enhanced software vulnerability detection using ChatGPT. *arXiv* **2023**, arXiv:2308.12697.
56. Wang, B.; Min, S.; Deng, X.; Shen, J.; Wu, Y.; Zettlemoyer, L.; Sun, H. Towards understanding chain-of-thought prompting. *arXiv* **2022**, arXiv:2212.10001.
57. Chen, Q.; Qin, L.; Liu, J.; Peng, D.; Guan, J.; Wang, P.; Hu, M.; Zhou, Y.; Gao, T.; Che, W. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv* **2025**, arXiv:2503.09567. [CrossRef]
58. Wang, Y.; Yu, Y.; Liang, J.; He, R. A comprehensive survey on trustworthiness in reasoning with large language models. *arXiv* **2025**, arXiv:2509.03871. [CrossRef]
59. McNemar, Q. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef] [PubMed]
60. McHugh, M.L. Interrater reliability: The kappa statistic. *Biochem. Med.* **2012**, *22*, 276–282. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.