EVALUATING LARGE LANGUAGE MODELS ON THE 2026 KOREAN CSAT MATHEMATICS EXAM: MEASURING MATHEMATICAL ABILITY IN A ZERO–DATA-LEAKAGE SETTING

Goun Pyeon[*], **Inbum Heo**[*], **Jeesu Jung**[*], **Taewook Hwang**[*], **Hyuk Namgoong**, **Hyein Seo**, **Yerim Han**, **Eunbin Kim**, **Hyeonseok Kang**, **Sangkeun Jung**[†]
Department of Computer Science & Engineering
Chungnam National University
{vusrhdns714, inbum10222, jisu.jung5, taewook5295, hyuk199, hyenee97, namu.rim2, eunbinkim777, dnfldjaak11, hugmanskj}@gmail.com

ABSTRACT

This study systematically evaluated the mathematical reasoning capabilities of Large Language Models (LLMs) using the 2026 Korean College Scholastic Ability Test (CSAT) Mathematics section, ensuring a completely **contamination-free** evaluation environment. To address data leakage issues in existing benchmarks, we digitized all 46 questions (22 common and 24 elective) within **two hours** of the exam's public release, eliminating any possibility of inclusion in model training data. We conducted comprehensive evaluations of 24 state-of-the-art LLMs across varying input modalities (Text-only, Image-only, Text+Figure) and prompt languages (Korean, English). The GPT-5 family models achieved perfect scores (100 points) under a limited set of language–modality configurations, while Grok 4, Qwen 3 235B, and Gemini 2.5 pro also scored above 97 points. Notably, gpt-oss-20B achieved 95.7 points despite its relatively small size, demonstrating high cost-effectiveness. Problem-specific analysis revealed Calculus as the weakest domain with significant performance degradation on 4-point high-difficulty problems. Text input consistently outperformed image input, while prompt language effects varied by model scale. In reasoning enhancement experiments with GPT-5 series, increased reasoning intensity improved performance (82.6→100 points) but quadrupled token usage and drastically reduced efficiency, suggesting that models with minimal reasoning may be more practical. This research contributes: (1) implementation of a completely unexposed evaluation environment, (2) a standardized digitization pipeline that converts human-targeted exam materials into LLM-ready evaluation data, and (3) a practical evaluation perspective integrating performance, cost, and time considerations. Detailed results and model comparisons are available at the 2026 Korean CSAT LLM Evaluation Leaderboard; https://isoft.cnu.ac.kr/csat2026/.

***K*eywords** Large Language Models (LLMs) · Mathematical Reasoning and Efficiency · Prompting Strategies · Contamination-Free Math Benchmarks

1 Introduction

The mathematical reasoning abilities of Large Language Models (LLMs) have become a central indicator for assessing a model's logical thinking and problem-solving skills. Because mathematical problems demand clear answers and structured reasoning, they are widely used as benchmarks for evaluating a model's true reasoning capabilities. However, with the rapid performance gains of recent models, traditional evaluation approaches have increasingly struggled to accurately distinguish an LLM's genuine mathematical competence.

[*]These authors contributed equally.
[†]Corresponding author.

In particular, given that LLMs are trained on vast web-scale corpora, it is difficult to verify whether a model has been exposed—directly or indirectly—to publicly available benchmark datasets or closely related materials [Min et al., 2022, Dodge et al., 2021]. Even when a benchmark itself is not included in training data, paraphrased, synthetic, or structurally transformed versions of benchmark problems may still have been used in training or inference.

Such **potential contamination** to benchmark data can distort measured performance, making it difficult to determine whether a model solves problems using generalized mathematical reasoning or merely reproduces patterns memorized during training. In other words, an LLM may perform well on a public benchmark simply because it has been implicitly optimized for those questions. Recent studies suggest that benchmark contamination may have occurred in the training data of major LLMs such as GPT-4 and Claude, raising the possibility that high scores may reflect *memorization or pattern replication* rather than true reasoning ability [Balloccu et al., 2024, Deng et al., 2024]. Thus, an evaluation dataset that is completely free from any risk of data leakage is essential for validating an LLM's genuine mathematical understanding and problem-solving ability.

In this context, the Mathematics section of the Korean College Scholastic Ability Test (CSAT) provides a uniquely leakage-free evaluation environment. The CSAT is a national-level exam created annually by the government, and because test items remain undisclosed until the exam ends, rapid digitization and evaluation immediately after release ensures that no LLM could have incorporated similar problems into its training data. Moreover, with 46 questions covering core domains such as *Probability and Statistics*, *Calculus*, and *Geometry*, the CSAT Mathematics exam enables comprehensive measurement of an LLM's practical mathematical proficiency across the entire secondary curriculum.

This study establishes a new evaluation framework that systematically assesses the mathematical reasoning capabilities of state-of-the-art LLMs under a fully contamination-free setting, using the Mathematics section of the 2026 CSAT. By digitizing and preprocessing all questions within two hours of their public release, we constructed a *Contamination-Free CSAT benchmark* that could not appear in the training data of any model. Beyond simple accuracy comparison, our framework varies input modality, prompting language, reasoning configuration, and solving time, enabling a multidimensional analysis of how LLMs interpret and solve real exam problems.

This work focuses on the following four research questions (RQs):

- **RQ1. What are the mathematical problem-solving abilities of LLMs on the 2026 CSAT?**
- **RQ2. Which types of CSAT Mathematics problems are most challenging for LLMs?**
- **RQ3. How does LLM mathematical performance vary across different input conditions?**
- **RQ4. What impact does varying the strength of reasoning have on model performance?**

Our results show that although several GPT-5 models achieved perfect scores under specific language–modality combinations, the GPT-5 family was the only group to reach such performance levels. In contrast, many other models exhibited notably poor performance on Calculus questions and on high-difficulty 4-point items, particularly short-answer problems.

Text-only input consistently outperformed image-based input, and increasing the reasoning strength of GPT-5 improved accuracy from 82.6 to 100 points, but with a 4–5× increase in token usage, sharply reducing efficiency. Notably, the small-scale gpt-oss-20B achieved 95.7 points at the lowest cost ($0.01), suggesting that model size is not a determining factor in mathematical performance.

The structure of this paper is as follows. Section 2 reviews prior work on math benchmarks and LLM evaluation. Section 3 analyzes the CSAT problem structure and introduces our CSAT-Math evaluation framework, including the data construction pipeline and prompt design. Section 4 describes the evaluated models and experimental setup. Section 5 presents experiments and results corresponding to the four research questions (RQ1–RQ4). Finally, Section 6 discusses key findings, Section 7 outlines the main limitations, and Section 8 concludes with future research directions.

2 Related Work

2.1 Math Benchmarks for LLMs

A wide range of benchmarks has recently been proposed to evaluate the mathematical reasoning abilities of large language models (LLMs). These benchmarks can be broadly categorized into **(1) benchmarks specifically constructed for LLM performance evaluation** and **(2) benchmarks based on real standardized exam problems**.

The first category—LLM-specific math benchmarks—consists of datasets designed to precisely measure a model's computational and reasoning abilities. GSM8K [Cobbe et al., 2021] evaluates basic numerical reasoning and short

chain-of-thought (CoT) capabilities using grade-school–level word problems. CHAMP [Mao et al., 2024] includes hints and intermediate reasoning steps to analyze how much a model depends on external information, while GSM-Plus [Li et al., 2024] introduces paraphrasing and numerical perturbation to test robustness under distribution shifts. More recently, multimodal benchmarks such as MathVista [Lu et al., 2024] and MATH-Vision [Wang et al., 2024] incorporate visual information—figures, diagrams, and mathematical notation—to evaluate vision–language integrated reasoning.

The second category evaluates LLMs using real exam problems. International standardized tests such as the SAT, GRE, AMC, AIME, and IMO are commonly used, and a representative math benchmark[3] for LLMs relies on AIME problems. Benchmarks such as MATH [Hendrycks et al., 2021], MathBench [Liu et al., 2024], OlymMATH [Sun et al., 2025], and U-MATH [Chernyshev et al., 2024] include problems drawn from real examinations and assess conceptual understanding and logical structuring. While these benchmark suites expand beyond simple accuracy—incorporating reasoning coherence and explanation quality—they remain predominantly focused on English-language, Western exam contexts.

In contrast, our work systematically evaluates LLM math performance in a non-English educational setting using the Korean CSAT Mathematics exam. The CSAT is a nationwide high-stakes university entrance exam taken annually by over 500,000 students and is regarded as a **national standard exam with high validity, reliability, and carefully balanced difficulty**. By using all 46 questions—22 from the common curriculum and 24 from elective subjects—we conduct a **comprehensive evaluation** covering geometry, calculus, and probability and statistics.

2.2 Data Leakage in LLM Evaluation

Data leakage poses a critical threat to the reliability of LLM performance evaluation and has been a central topic of discussion in math and language benchmark research.

Xu et al. [2024] showed that major math benchmarks such as GSM8K and MATH appear in the training data of modern LLMs, reporting contamination across 31 models. Hidayat et al. [2025] quantified contamination rates in large-scale benchmarks including MMLU and HellaSwag, while Balunović et al. [2025] demonstrated that performance drops substantially when LLMs are evaluated on MathArena, a private mathematics competition dataset unavailable during training. Together, these findings suggest that widely used benchmarks may fail to measure true reasoning ability, as models may exploit memorized patterns rather than demonstrate genuine generalization.

To eliminate this risk, our study **collects and processes the 2026 CSAT Mathematics exam immediately after public release** and conducts model evaluation right away, ensuring that no LLM could have been exposed to the data beforehand. This procedure guarantees a fully contamination-free setting, **blocking any possibility of prior contamination and enabling faithful measurement of mathematical reasoning using real, previously unseen exam problems**.

2.3 Prior Work on Input Modality and Prompting Language

LLM performance is influenced not only by model architecture but also by **input modality** and **prompting language**. Early LLMs were optimized for text-only input, but recent advances have produced **multimodal LLMs** capable of jointly processing images, diagrams, and mathematical notation. This has driven interest in evaluating models on visually grounded mathematics tasks [Lu et al., 2024, Wang et al., 2024].

Prompting language is another key factor. Research on multilingual prompting shows that the chain-of-thought structure, reasoning trajectory, and solution strategy can vary depending on the language in which the problem is presented. Several studies report that English prompts often provide greater stability and accuracy [Park et al., 2025, Zhou et al., 2025].

Building on this literature, we analyze how input modality (text, image, text+visual information) and prompting language (Korean, English) affect model performance in a realistic exam setting. By crossing these factors, we evaluate and compare LLM mathematical reasoning from multiple perspectives.

2.4 Prior Work on LLM Reasoning and Reasoning Evaluation

Research on **reasoning** in LLMs focuses on how to elicit, regulate, and evaluate the **reasoning process** a model performs before generating a final answer. Zero-shot Chain-of-Thought (CoT) [Kojima et al., 2022] demonstrated that simple prompts such as "Let's think step by step" can induce multi-step reasoning without fine-tuning. Wang et al.

[3] `https://artificialanalysis.ai/evaluations/aime-2025`

[2022] proposed generating multiple reasoning paths in parallel and selecting the final answer using majority voting, improving stability and consistency.

A growing body of work examines how to scale test-time reasoning. Snell et al. [2024] showed that increasing reasoning depth and compute—known as **test-time scaling**—can yield stronger performance. Zhou et al. [2025] visualized various reasoning types and cognitive trajectories, enabling qualitative assessment of reasoning quality. These studies collectively highlight that the relationship between reasoning length and accuracy is not linear: excessively long reasoning may accumulate errors and degrade performance.

Our work quantitatively evaluates these insights in a real exam setting. By systematically varying the reasoning mode (e.g., `reasoning_effort`) on the same CSAT problems, we analyze the *trade-off* between reasoning strength, performance, and computational cost.

3 Korean CSAT-Math Evaluation Framework for LLMs

3.1 Overview of the CSAT

The Korean College Scholastic Ability Test (CSAT; Korean: 수능) is a nationally administered, high-stakes standardized examination produced by the Korea Institute for Curriculum and Evaluation (KICE). It serves as the most authoritative assessment used for university admissions in Korea and evaluates whether students possess the academic abilities required for higher education[4].

The CSAT is administered once per year and taken by roughly half a million students nationwide—for example, 554,174 examinees in the 2026 CSAT. Because of its societal significance, the exam has substantial nationwide impact. For example, during the 2026 CSAT held on November 13, 2025, the morning commute for office workers was delayed to support test-takers, and all aircraft takeoffs and landings were suspended during the English listening section to minimize noise[5].

Due to its importance, the CSAT is constructed under extremely strict security protocols to prevent leakage. All exam writers and reviewers are isolated in a confidential facility beginning approximately 40 days before the exam, with complete prohibition of external communication and electronic devices[6]. Because the problem-writing, review, printing, and distribution processes are fully secured end-to-end, the CSAT is widely considered a **near-impossible-to-leak examination**.

The CSAT consists of five subject areas: Korean, Mathematics, English, Korean History, and elective Social/Science/Vocational studies. This work focuses exclusively on the **Mathematics section**.

Since 2022, the CSAT Mathematics section has consisted of 22 common-subject questions (Math I and Math II) and 8 elective questions selected from Probability and Statistics, Calculus, or Geometry, for a total of 30 questions to be solved in 100 minutes.

In this study, to evaluate overall mathematical competence and cross-subject reasoning, we include **all 46 questions**—22 common-subject problems and all 24 elective-subject problems from the three elective domains—regardless of the single-subject choice that human test-takers make.

Table 1 summarizes the complete structure, problem types, and scoring scheme.

- **Common Subject (22 problems)**: Math I (11 problems) and Math II (11 problems), taken by all test-takers.
- **Elective Subject (8 problems)**: Students select one among Probability and Statistics, Calculus, and Geometry.
- **Question Types**: Multiple-choice and short-answer.
  - **Short-answer**: Students directly provide an integer between 0 and 999.
  - **Multiple-choice**: Five answer options (1–5) are provided.
- **Scoring**: Each problem is worth 2, 3, or 4 points depending on required reasoning complexity.

According to the official 2026 CSAT Mathematics curriculum guide[7], each subject covers the following areas:

- **Common Subjects**

[4] `https://suneung.re.kr/sub/info.do?m=0101&s=suneung`
[5] `https://www.yna.co.kr/view/AKR20251111058900003`
[6] `https://www.news1.kr/society/education/5973136`
[7] `https://www.suneung.re.kr/boardCnts/fileDown.do?fileSeq=ead4803d0df8cda516056b38fef4c7ae`

Table 1: Structure and Question Types of the CSAT Mathematics Section

		Common	Elective			Total
Subject		Math I + Math II	Probability	Calculus	Geometry	
Num. of Questions		11 + 11	8	8	8	46
Type	Multiple-choice	15	6	6	6	33
	Short-answer	7	2	2	2	13
Score	2-point problems	2	1	1	1	5
	3-point problems	6	4	4	4	18
	4-point problems	14	3	3	3	23
Total Score		74	26	26	26	152

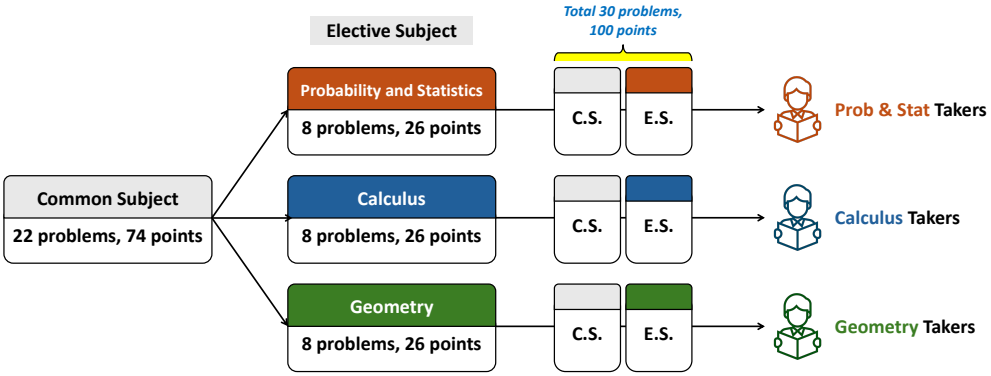


Figure 1: Overall structure of the CSAT Mathematics Section (Common (C.S.) vs. Elective Subjects (E.S.))

  - **Math I**: Exponential and logarithmic functions, trigonometric functions, sequences
  - **Math II**: Limits and continuity, differentiation, integration
- **Elective Subjects**
  - **Probability and Statistics**: Counting, probability, statistical inference
  - **Calculus**: Limits of sequences, differential calculus, integral calculus
  - **Geometry**: Conic sections, plane vectors, three-dimensional geometry and coordinates

Because the CSAT Mathematics exam has a systematically designed structure—including well-defined scope, problem types, and difficulty—it is particularly suited for evaluating LLM mathematical reasoning. The exam follows a consistent progression from common to elective subjects, as visualized in Figure 1.

3.2 Data Construction Pipeline

Our evaluation framework is designed to **completely eliminate** the possibility of prior contamination during model training. To achieve this, we constructed the evaluation dataset and began model evaluation within the first two hours after the official release of the exam. A total of seven researchers (six annotators and one reviewer) contributed to the manual dataset construction, and eight participated in the LLM evaluation and monitoring process. Figure 2 illustrates the full pipeline executed immediately after exam release.

The dataset construction process proceeds as follows:

1. Annotators receive the Digitalized CSAT-Math template and their assigned question IDs.
2. Immediately after the official exam release[8], each annotator extracts all required information for their assigned questions.
3. A reviewer aggregates all entries and checks for errors and consistency.
4. A unified Digitalized CSAT-Math dataset is finalized.

[8]Released at 14:10 KST via the KICE website: `https://cdn2.kice.re.kr/suneung-26/index.html`

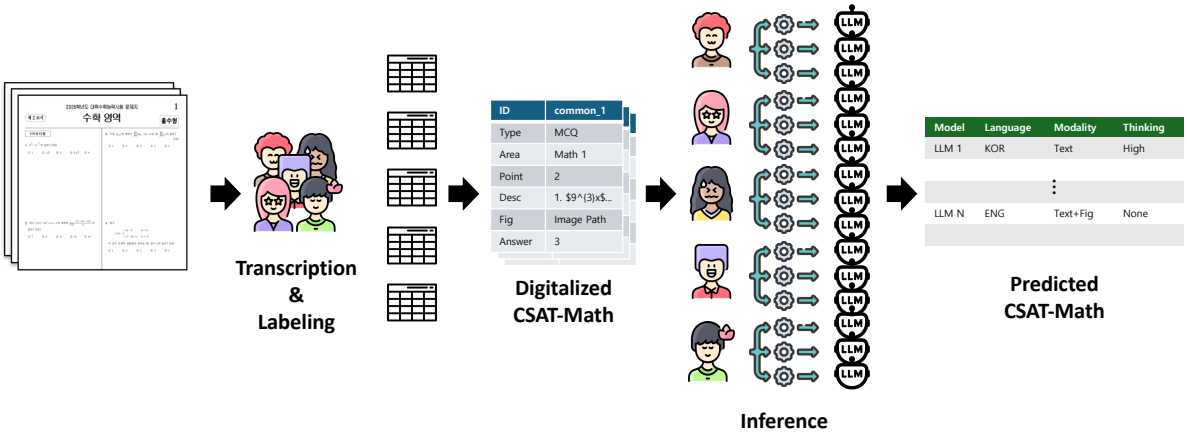


Figure 2: Manual dataset construction (transcription and labeling) and LLM evaluation pipeline

3.3 Structure of the 2026 CSAT Math Question Paper

The actual question booklet distributed to students consists of:

- Subject designation: Common and elective subjects (Probability, Calculus, Geometry)
- Question type header: Displayed at the start of each type section
- Problem statement: Question description with problem number and score
- Scoring: 2-, 3-, or 4-point values
- Visual materials: Diagrams, graphs, or figures accompanying the problem
- Tables: Used in a small number of questions (e.g., z-score tables in Probability/Statistics)
- Answer choices: Provided for multiple-choice questions (1–5)

3.3.1 Digitalized CSAT-Math Data Format

Annotators input structured information into the Digitalized CSAT-Math dataset as follows:

- `prob_id`: Unique problem ID. Example: `2026_odd_common_11` refers to the 11th common-subject problem in the odd-form 2026 exam; `2026_odd_probability_30` refers to the 30th problem in the Probability and Statistics section.
- `prob_type`: Question type (multiple-choice or short-answer)
- `prob_area`: Subject area (Math I, Math II, Probability, Calculus, Geometry)
- `prob_point`: Score value (2, 3, or 4 points)
- `prob_desc`: Problem text converted into Markdown + LaTeX
- `prob_img_path`: Cropped image of the full problem
- `prob_fig_img_path`: Cropped image of figures or diagrams only
- `answer`: Correct answer (1–5 for multiple-choice; integer 0–999 for short-answer)

From 2022 to 2026, `prob_id`, `prob_type`, and `prob_point` remain fixed for each problem index. For example, problems 1–22 are always common-subject questions, and problems 23–30 always correspond to the three elective subject sets. Therefore, annotators were instructed to input only the five variable fields: `prob_area`, `prob_desc`, `prob_img_path`, `prob_fig_img_path`, and `answer`.

3.4 Digitalized CSAT-Math Dataset Construction

The goal of the dataset construction process is twofold: (1) to **preserve the original structure and mathematical expressions of the exam problems as faithfully as possible**, and (2) to **standardize the data into a consistent format suitable for direct LLM input**.

A total of seven annotators participated in building the Digitalized CSAT-Math dataset. Among the 46 questions, four annotators each handled 9 questions and one annotator handled 10 questions, producing the fields `prob_desc`, `answer`, and `prob_area`. Additionally, one annotator exclusively extracted all problem and figure images.

3.4.1 Text Construction for Each Problem

To enable LLM-based evaluation using PDF CSAT problems, all questions were fully transcribed into LLM-friendly text. Each problem's stem, choices, score, tables, and mathematical expressions were encoded into the `prob_desc` field of the Digitalized CSAT-Math format. The `prob_desc` field represents a textual reconstruction of the "human-readable problem sheet."

Extraction of Structured Content Using Markdown Markdown provides a lightweight syntax for expressing structured document elements using only plain text. For example, ">" represents a quotation block, and tables can be written using the syntax "`| a | b |`".

CSAT mathematics problems contain various structural elements such as boxed passages (Figure 3 f, i), multiple-choice options (Figure 3 b, d), and occasional tables (Figure 3 k). When naively converted to plain text, such structures often lose their meaning or readability. By contrast, Markdown naturally preserves these structures without information loss.

General paragraphs were written as plain text (Figure 3 a, c, h), while boxed "보기" sections were represented as block quotes using ">" (Figure 3 f, i). Using Markdown allows the reconstructed text to closely resemble the original layout, and can be visually rendered to match the PDF problem sheet.

Mathematical Expressions Using LaTeX All mathematical expressions were converted into LaTeX, ensuring clear boundaries between text and math and allowing LLMs to interpret expressions accurately. Inline formulas were written as `$...$` (Figure 3 f, g, j), and block-level expressions were written as `$$...$$` (Figure 3 e).

We reproduced the printed formulas in the CSAT as closely as possible: non-italic text (e.g., P) was expressed using `\mathrm{}`, vectors and segment lengths were written using `\vec{AB}` and `\overline{AB}` respectively, and operators such as limits or summations used `\limits` to ensure upper/lower bounds appear above and below as in the original problem sheet.

Representation of Problem Components (Number, Score, Answer Choices) Problem numbers, scoring labels (e.g., "[4점]"), and answer choices were transcribed exactly as printed. Multiple-choice options were preserved using original circled numerals (①–⑤), e.g., "① 2 ② 3 ③ 4 ④ 5 ⑤ 6". This ensured that the Digitalized CSAT-Math text closely mirrored the actual problem layout.

3.4.2 Metadata Construction for Each Problem

We define **metadata** as information that is required for evaluation and analysis but is not explicitly visible to LLMs in the original exam sheet. This includes the true *subject area* of each problem and its *correct answer.*

Actual Subject Area per Problem As described in Section 3.1, CSAT Mathematics consists of Math I, Math II, and one of three elective subjects. However, within the 1–22 common-subject questions, the subject source (Math I vs. Math II) is not indicated in the problem sheet and must be manually identified.

Because this information is not officially provided, we manually annotated each problem in the 1–22 range as belonging to Math I or Math II, constructing the `prob_area` field accordingly.

Ground-Truth Answers The `answer` field holds the correct answer for each problem. Multiple-choice answers were stored as integers corresponding to their options (e.g., ① → `1`, ⑤ → `5`). Short-answer problems were recorded as integers within the 0–999 range. Expressing all answers in a unified numerical form enables consistent scoring across both multiple-choice and short-answer problems.

3.4.3 Image Data Construction

Image Extraction Many CSAT math questions contain diagrams or visual elements. In the 2026 exam, 8 of the 46 questions included such visual components. To analyze the effect of visual information on LLM mathematical reasoning, we extracted two types of images from the original PDF:

- **Problem Image**: Full problem region (`prob_img_path`)

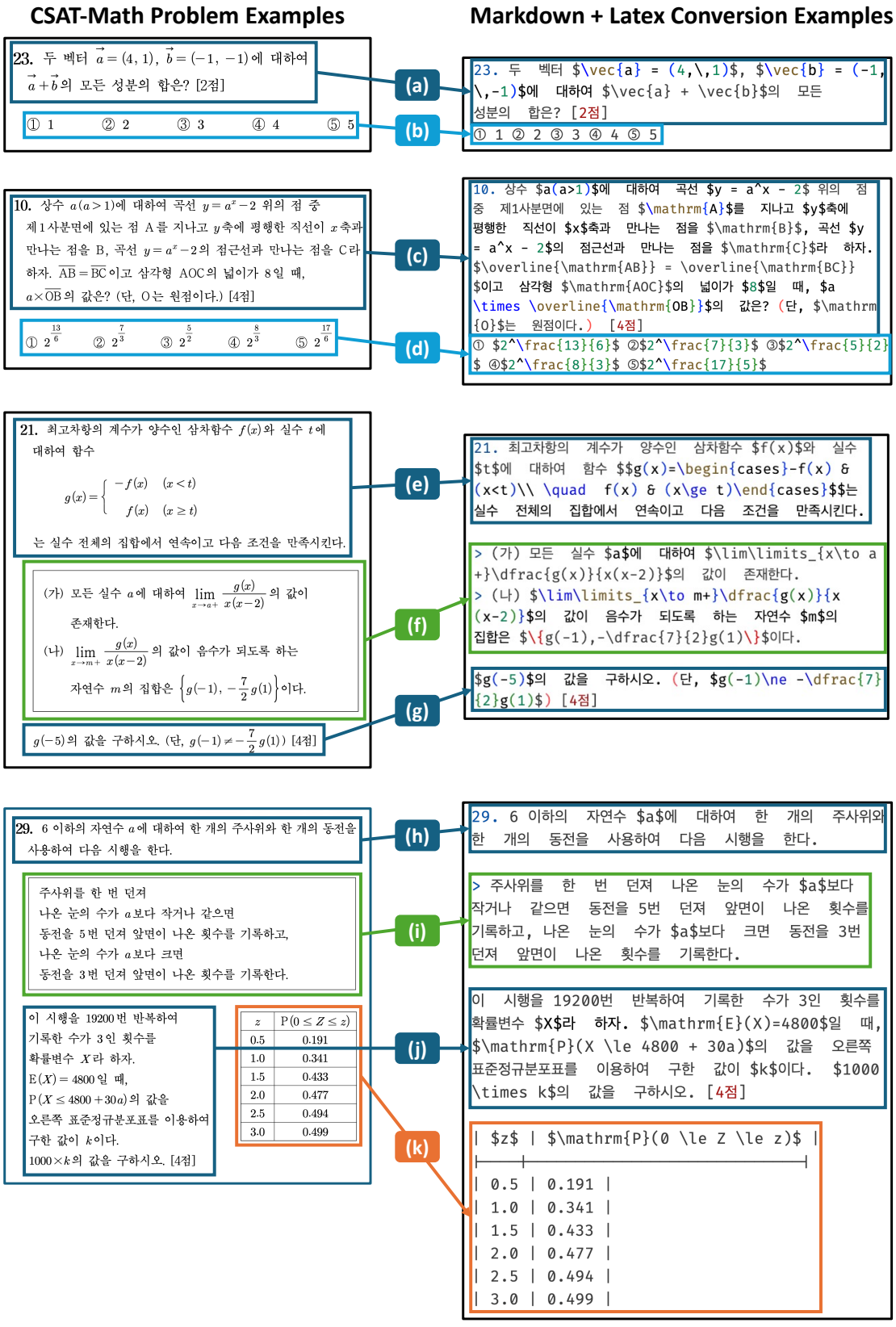


Figure 3: Example of a real CSAT problem and its Digitalized CSAT-Math textual representation

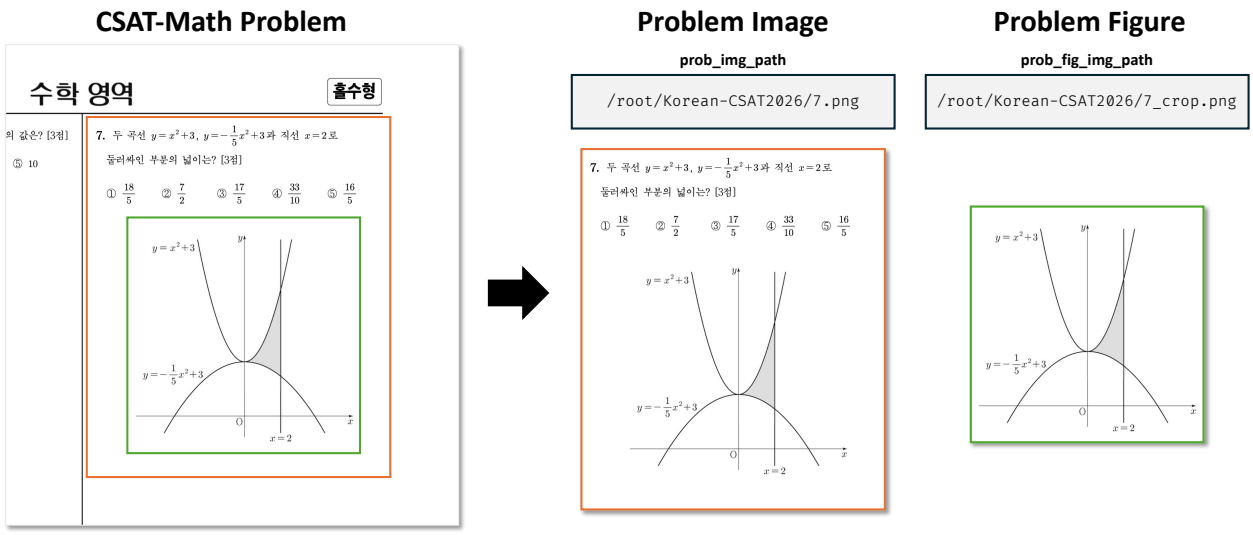


Figure 4: Examples of extracted CSAT problem images

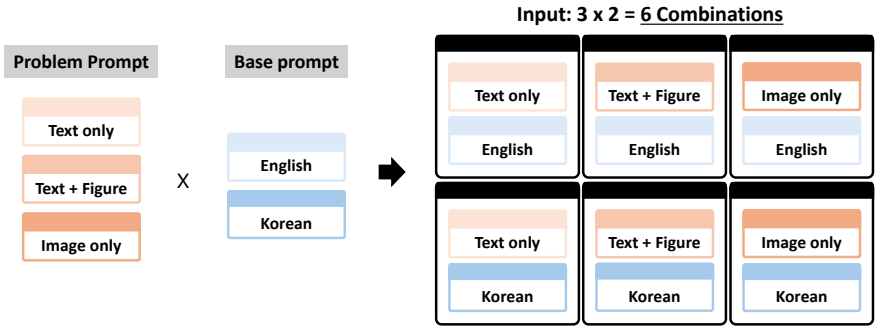


Figure 5: Example of prompt composition

- **Problem Figure**: Only diagrams/graphs/visual elements (`prob_fig_img_path`)

Figure 4 shows examples of these extracted image components.

3.4.4 Quality Assurance

Annotators and reviewers rendered each problem using Markdown editors (Obsidian[9] or StackEdit[10]) to verify visual alignment with the original PDF problem sheet. They checked Markdown+LaTeXrendering accuracy—including formula formatting, line breaks, symbols, and layout. A final pass by the supervisor ensured consistency and correctness across all entries.

3.5 Prompt Template for Problem Solving

LLM prompts consist of two components: (1) the **problem**, containing the CSAT math content, and (2) the **instruction**, which guides the model on how to solve the problem in a CSAT-appropriate manner.

To evaluate both multimodal and multilingual capabilities, we created multiple combinations of problem input modalities and instruction languages. There are three input modalities and two instruction-language settings, resulting in six total prompt conditions. Figure 5 illustrates these combinations.

Input Modalities Because CSAT questions may include both textual and visual elements, we evaluate LLMs under three input formats:

[9] `https://obsidian.md/`
[10] `https://stackedit.io/`

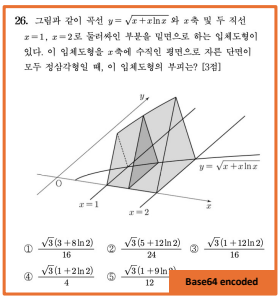
Input Modality

Problem Prompt – Text only

26. 그림과 같이 곡선 $y = \sqrt{x + x\ln x}$와 x축 및 두 직선 $x = 1$, $x = 2$로 둘러싸인 부분을 밑면으로 하는 입체도형이 있다. 이 입체도형을 x축에 수직인 평면으로 자른 단면이 모두 정삼각형일 때, 이 입체도형의 부피는? [3점]

① $\displaystyle \frac{\sqrt{3}(3 + 8\ln 2)}{16}$ ② $\displaystyle \frac{\sqrt{3}(5 + 12\ln 2)}{24}$ ③ $\displaystyle \frac{\sqrt{3}(1 + 12\ln 2)}{16}$ ④ $\displaystyle \frac{\sqrt{3}(1 + 2\ln 2)}{4}$ ⑤ $\displaystyle \frac{\sqrt{3}(1 + 9\ln 2)}{12}$

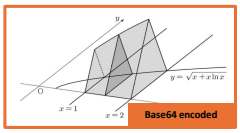
Problem Prompt – Image only



Problem Prompt – Text + Figure

26. 그림과 같이 곡선 $y = \sqrt{x + x\ln x}$와 x축 및 두 직선 $x = 1$, $x = 2$로 둘러싸인 부분을 밑면으로 하는 입체도형이 있다. 이 입체도형을 x축에 수직인 평면으로 자른 단면이 모두 정삼각형일 때, 이 입체도형의 부피는? [3점]

① $\displaystyle \frac{\sqrt{3}(3 + 8\ln 2)}{16}$ ② $\displaystyle \frac{\sqrt{3}(5 + 12\ln 2)}{24}$ ③ $\displaystyle \frac{\sqrt{3}(1 + 12\ln 2)}{16}$ ④ $\displaystyle \frac{\sqrt{3}(1 + 2\ln 2)}{4}$ ⑤ $\displaystyle \frac{\sqrt{3}(1 + 9\ln 2)}{12}$



Prompt Language

Base Prompt – English

Solve this question.

Do these steps and SHOW your work:
1) Translate the problem into English.
2) Solve the problem with clear reasoning and calculations.
3) State the final answer.

MANDATORY: Write the FINAL answer on the LAST line in this exact format:
\\ boxed{answer}
Nothing after the boxed answer.

Type-specific output rules:
- Multiple-choice (five options): answer MUST be ONLY an integer 1–5.
- Short-answer: answer MUST be ONLY an integer in [0, 999], no units, no leading zeros.

Examples:
- MCQ → \\ boxed{3}
- Short-answer → \\ boxed{256}

Problem type: 5지선다형
Problem ID: 2026_odd_common_11

Base Prompt – Korean

다음 수학 문제를 풀어주세요.

아래 순서로, 과정을 반드시 보여주며 작성하세요:
1) 문제를 한국어로 정확히 이해하고 핵심 조건을 정리합니다.
2) 논리적 추론과 계산 과정을 단계별로 서술해 풀이합니다.
3) 최종 정답을 제시합니다.

필수: 마지막 줄에 아래 형식으로 '딱 한 줄'만 출력하세요:
\\boxed{{answer}}
박스 이후에는 아무 내용도 쓰지 않습니다.

문항 유형별 규칙:
- 객관식(5지선다): 정답은 1~5 중 하나의 '정수'만.
- 단답형: 정답은 0~999 범위의 '정수'만(단위/불필요한 0 금지).

예시:
- 객관식 → \\boxed{3}
- 단답형 → \\boxed{256}

Problem type: 5지선다형
Problem ID: 2026_odd_common_11

Example 1

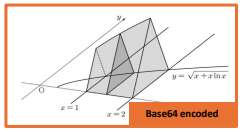
Problem prompt: Text+Figure

Base prompt: Korean

Prompt

26. 그림과 같이 곡선 $y = \sqrt{x + x\ln x}$와 x축 및 두 직선 $x = 1$, $x = 2$로 둘러싸인 부분을 밑면으로 하는 입체도형이 있다. 이 입체도형을 x축에 수직인 평면으로 자른 단면이 모두 정삼각형일 때, 이 입체도형의 부피는? [3점]

① $\displaystyle \frac{\sqrt{3}(3 + 8\ln 2)}{16}$ ② $\displaystyle \frac{\sqrt{3}(5 + 12\ln 2)}{24}$ ③ $\displaystyle \frac{\sqrt{3}(1 + 12\ln 2)}{16}$ ④ $\displaystyle \frac{\sqrt{3}(1 + 2\ln 2)}{4}$ ⑤ $\displaystyle \frac{\sqrt{3}(1 + 9\ln 2)}{12}$



다음 수학 문제를 풀어주세요.

아래 순서로, 과정을 반드시 보여주며 작성하세요:
1) 문제를 한국어로 정확히 이해하고 핵심 조건을 정리합니다.
2) 논리적 추론과 계산 과정을 단계별로 서술해 풀이합니다.
3) 최종 정답을 제시합니다.

필수: 마지막 줄에 아래 형식으로 '딱 한 줄'만 출력하세요:
\\boxed{{answer}}
박스 이후에는 아무 내용도 쓰지 않습니다.

문항 유형별 규칙:
- 객관식(5지선다): 정답은 1~5 중 하나의 '정수'만.
- 단답형: 정답은 0~999 범위의 '정수'만(단위/불필요한 0 금지).

예시:
- 객관식 → \\boxed{3}
- 단답형 → \\boxed{256}

Problem type: 5지선다형
Problem ID: 2026_odd_common_11

Example 2

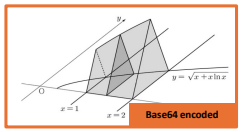
Problem prompt: Text+Figure

Base prompt: English

Prompt

26. 그림과 같이 곡선 $y = \sqrt{x + x\ln x}$와 x축 및 두 직선 $x = 1$, $x = 2$로 둘러싸인 부분을 밑면으로 하는 입체도형이 있다. 이 입체도형을 x축에 수직인 평면으로 자른 단면이 모두 정삼각형일 때, 이 입체도형의 부피는? [3점]

① $\displaystyle \frac{\sqrt{3}(3 + 8\ln 2)}{16}$ ② $\displaystyle \frac{\sqrt{3}(5 + 12\ln 2)}{24}$ ③ $\displaystyle \frac{\sqrt{3}(1 + 12\ln 2)}{16}$ ④ $\displaystyle \frac{\sqrt{3}(1 + 2\ln 2)}{4}$ ⑤ $\displaystyle \frac{\sqrt{3}(1 + 9\ln 2)}{12}$



Solve this question.

Do these steps and SHOW your work:
1) Translate the problem into English.
2) Solve the problem with clear reasoning and calculations.
3) State the final answer.

MANDATORY: Write the FINAL answer on the LAST line in this exact format:
\\ boxed{answer}
Nothing after the boxed answer.

Type-specific output rules:
- Multiple-choice (five options): answer MUST be ONLY an integer 1–5.
- Short-answer: answer MUST be ONLY an integer in [0, 999], no units, no leading zeros.

Examples:
- MCQ → \\ boxed{3}
- Short-answer → \\ boxed{256}

Problem type: 5지선다형
Problem ID: 2026_odd_common_11

Figure 6: Example of a full prompt

- **Text only**: The problem is provided solely as text (`prob_desc`). All diagrams or figures are omitted. This modality is supported by all models.
- **Image only**: The model receives the full problem as an image (Problem Image) along with the instruction. This setting mirrors actual test-taking conditions and evaluates the model's OCR and visual reasoning capabilities. Text-only LLMs are excluded from this condition.
- **Text + Figure**: The textual portion is provided as `prob_desc`, while diagrams or graphs are separately provided as images (Problem Figure). This hybrid setting requires both text understanding and visual interpretation. Text-only models are excluded.

Prompting Language To analyze multilingual prompting behavior, we use two instruction-language conditions:

- **Korean**: Both the problem and the instruction are presented in Korean.
- **English**: The problem text remains in Korean, but the instruction is given in English.

Combining three modalities and two languages yields six conditions (Figure 5). Text-only models are evaluated under the two text-based conditions, while multimodal models are evaluated under all six. Examples of full prompt configurations are shown in Figure 6.

4 CSAT-Math LLM Experiment Design

In this section, we describe the overall experimental design. Specifically, we (1) detail the list of LLMs used in the evaluation and their execution environment, and (2) explain the experimental goals and procedures aligned with the four core research questions (RQs).

4.1 Model List

We target high-performing LLMs capable of solving CSAT-level math problems, while also considering practical deployability. To this end, we restrict our evaluation to models satisfying the following criteria:

- **Performance**: To pre-screen mathematical capabilities, we only consider the 24 models highlighted on the AIME 2025 benchmark homepage[11] as of October 5, 2025. This allows us to compare detailed behaviors on the CSAT among models that already demonstrate strong performance and recognition on a challenging competition-style math benchmark.
- **Accessibility**: For experimental convenience and reproducibility, we only include models that were directly accessible via OpenRouter as of October 5, 2025. Using a unified API interface simplifies our evaluation pipeline and enables future researchers to reproduce or extend our setup without changing the underlying infrastructure.
- **Release time**: To maintain a **zero-leakage** evaluation environment, we only include models that were released **before** the 2026 CSAT date (November 13, 2025). Models released or updated after the exam date are excluded, as they may potentially have been trained or tuned on post-exam content.
- **Diversity**: To avoid bias toward specific model families, we balance **proprietary models** (e.g., GPT, Claude, Gemini) and **open-weight models** (e.g., Qwen, Llama, DeepSeek). We also include models with varying parameter scales (from billions to hundreds of billions) and architectural designs (standard LLMs vs. reasoning-focused variants) to analyze performance differences as a function of family and size under a shared evaluation setting.

Table 2 summarizes the final set of models, including provider, API pricing, parameter scale, and whether a built-in reasoning/thinking mode is supported.

4.2 Configuring Model Reasoning

Modern LLMs increasingly move beyond simple pattern matching and incorporate built-in **reasoning** mechanisms that analyze problem structure and generate intermediate thoughts. In this work, we define reasoning as **"the internal thinking or multi-step inference process a model performs before producing a final answer"**. Typically, such reasoning processes consume separate reasoning tokens, distinct from the visible output tokens.

[11] `https://artificialanalysis.ai/evaluations/aime-2025`

Table 2: List of LLMs, providers, prices, model scales, and Reasoning/Thinking support. "1M Input" and "1M Output" denote the price per 1 million input/output tokens, respectively, as specified on OpenRouter by each model provider. Model scale is reported in terms of active parameters; when not publicly available, we estimate it based on prior documentation and reported performance (marked with $\sim^*$). Reasoning support is marked with ✓ if available and "–" otherwise.

Type		Model	Model Provider	1M Input	1M Output	Scale	Reasoning Support
Closed	GPT	GPT-5 Codex	openai	$1.25	$10.00	∼635B*	–
		GPT-5	openai	$1.25	$10.00	∼330B*	✓
		GPT-5-mini	openai	$0.25	$2.00	∼27B*	✓
		GPT-5-nano	openai	$0.05	$0.40	∼15B*	✓
	Grok	Grok 4	xai	$6.00	$30.00	∼213B*	✓
		Grok 4 Fast	xai	$0.40	$1.00	∼40B*	–
	Claude	Claude 4.5 Sonnet	anthropic	$6.00	$22.50	∼100B*	✓
		Claude 4.5 Haiku	anthropic	$1.00	$5.00	∼30B*	✓
		Claude 4.1 Opus	anthropic	$15.00	$75.00	∼400B*	✓
	Gemini	Gemini 2.5 Pro	google-vertex/global	$2.50	$15.00	∼128B*	✓
		Gemini 2.5 Flash	google-vertex/global	$0.30	2.50	∼27B*	–
Open	GPT	gpt-oss 20B	hyperbolic	$0.04	$0.04	3.6B	✓
	Qwen	Qwen3 235B A22B	siliconflow/fp8	$0.30	$1.50	22B	–
		Qwen3 235B A22B Thinking 2507	siliconflow/fp8	$0.45	$3.50	22B	✓
	Deepseek	Deepseek R1 0528	novita/fp8	$0.27	$0.41	∼37B*	✓
		Deepseek V3.2 Exp	novita/fp8	$0.70	$2.50	∼37B*	–
	GLM	GLM 4.6	z-ai	$0.60	$2.20	32B	–
	Magistral	Magistral Medium 2506	mistral	$2.00	$5.00	45B	–
		Magistral Medium 2506 Thinking	mistral	$2.00	$5.00	45B	✓
	MiniMax	MiniMax M2	minimax	$0.30	$1.20	∼10B*	–
	Llama	Llama 4 Maverick	friendli	$0.20	$0.60	17B	–
		Llama 3.3 Nemotron Super 49B V1.5	deepinfra/fp8	$0.10	$0.40	49B	–
	Kimi	Kimi K2 0905	moonshotai	$0.60	$2.50	32B	–
		Kimi K2 0905 Thinking	moonshotai	$0.60	$2.50	32B	✓

While both proprietary and open-weight models have begun to expose some reasoning functionality, these are often available only through dedicated "reasoning models" or with limited, coarse-grained options. Fine-grained control over reasoning strength is rarely supported. Therefore, for systematic comparison, we conduct a dedicated reasoning study only on GPT-5, which allows us to control both reasoning strength and explanation length.

GPT-5 exposes two key configuration knobs: (1) `Reasoning_Effort`: internal reasoning strength (amount of reasoning tokens), and (2) `Text_Verbosity`: length of the final explanation. These two parameters are independent: for example, one can use high reasoning effort with a concise explanation (high effort + low verbosity), or minimal reasoning with a long explanation (minimal effort + high verbosity).

Reasoning_Effort GPT-5 provides four discrete levels for internal reasoning:

- `minimal`: minimizes reasoning tokens; fastest responses
- `low`: basic reasoning; sufficient for simpler problems
- `medium` (default): balanced reasoning; stable for typical math problems
- `high`: deep reasoning; suitable for difficult, complex problems

Text_Verbosity `Text_Verbosity` controls the length of the final response, independently of `Reasoning_Effort`:

- `low`: concise answers; may omit detailed reasoning steps
- `medium` (default): standard explanatory length
- `high`: detailed, extended explanations

`Reasoning_Effort` × `Text_Verbosity` **Combinations** Combining the four effort levels with the three verbosity levels yields 12 distinct configurations. Table 3 summarizes their qualitative characteristics. We treat each configuration as a separate experimental condition and compare their behavior on the CSAT.

Table 3: Qualitative characteristics of GPT-5 Reasoning Effort × Text Verbosity combinations

Effort	Verbosity	Characteristics
minimal	low	Minimal reasoning × short output
	medium	Minimal reasoning × medium-length output
	high	Minimal reasoning × long output
low	low	Low reasoning × short output
	medium	Low reasoning × medium-length output
	high	Low reasoning × long output
medium	low	Medium reasoning × short output
	medium	Medium reasoning × medium-length output (default)
	high	Medium reasoning × long output
high	low	High reasoning × short output
	medium	High reasoning × medium-length output
	high	High reasoning ×long output

4.3 API and Runtime Environment

All experiments were conducted via the OpenRouter API[12], using the following fixed hyperparameters for consistent evaluation:

- **Temperature**: 0.0
  Temperature controls the randomness in token sampling. Higher values increase diversity in generated outputs, while values closer to 0 bias the model toward the most likely tokens, yielding more deterministic behavior. Since our primary goal is to analyze performance differences across input modalities (Text-only, Image-only, Text+Figure) and prompting languages (Korean, English), we fix Temperature at 0.0 to minimize randomness-induced variance.
- **Top-p**: 0.0
  Top-p controls nucleus sampling by restricting candidate tokens to a cumulative probability mass of p. Larger values introduce more variability; values near 0 strongly favor the most probable token. We set Top-p to 0.0 to eliminate sampling variability and ensure deterministic outputs.
- **Max tokens**: Unconstrained, up to each model's maximum setting.
- **Retry policy**: On API failure, we retry up to 5 times with an exponential backoff strategy[13] (e.g., 5s → 10s → 20s → 40s . . .), ensuring robust completion of large-scale experiments.
- **Timeout**: None.

Experiments were performed by eight researchers using personal PCs and servers. The environment is as follows:

- **OS**: Ubuntu 22.04 LTS or Windows 10
- **Language/runtime**: Python 3.10
- **Key libraries**:
  - `openai==2.8.0`: OpenRouter-compatible API calls and response handling
  - `pandas==2.3.3`: Structuring results and generating leaderboards
- **Parallelization**: Up to 16 processes in parallel. Each process is run via a single Python command, solving all 46 questions sequentially for a specific model–input-condition pair, orchestrated via shell scripts.

4.4 LLM Response Logging and Predicted CSAT-Math Format

The problem-solving script automatically logs all information at the question level for every API call. Logged fields include:

[12] `https://openrouter.ai`
[13] `https://en.wikipedia.org/wiki/Exponential_backoff`

- Full API request
- Full model response
- Extracted final answer
- Actual response time per problem (latency, KST)
- API token usage (input / output / total)
- Error type and number of retries (if any)

The **Predicted CSAT-Math format** is defined as follows:

- `prob_id`: Unique problem ID
- `prob_type`: Problem type [multiple-choice or short-answer]
- `prob_area`: Subject area [Math I, Math II, Probability and Statistics, Calculus, Geometry]
- `answer`: Ground-truth answer [①–⑤ for multiple-choice, 0–999 integer for short-answer]
- `model_index`: Internal model index
- `model_id`: Model identifier for OpenRouter requests (e.g., `openai/gpt-5-codex`)
- `provider`: Serving provider (e.g., `novita/fp8`), which can affect quantization and pricing
- `content_mode`: Input modality [`text_only`, `image_only`, or `text_fig`]
- `language`: Prompt language [`ko` (Korean) or `en` (English)]
- `variant_tag`: Tag used for naming log files
- `start_time_kst`: Start time (KST) of problem solving (e.g., 2025-11-13 15:52:28.894)
- `end_time_kst`: End time (KST) of problem solving (e.g., 2025-11-13 15:56:20.391)
- `model_answer`: Parsed answer extracted from the model output. The instruction requires that the final answer be enclosed in `\boxed{}`, and we parse the content inside the last `\boxed{}`.
- `elapsed_sec`: Time taken to solve the problem, computed from `start_time_kst` and `end_time_kst` (in seconds with millisecond precision)
- `prompt_tokens`: Number of input tokens used for the prompt
- `completion_tokens`: Number of tokens generated by the model
- `total_tokens`: Sum of input and output tokens, representing total token usage per problem
- `attempt_count`: Number of attempts; 1 if successful on the first try, 0 if failed due to unrecovered errors
- `error`: Logged error type, if any
- `request_body`: Additional options passed in the model call (e.g., `"Reasoning_effort": "high", "text_verbosity": "low"`)

Using this logs, we analyze correlations between performance and latency, modality-specific behavior, and stability across model families and sizes.

4.5 Goals of the CSAT-Math Experiments

To answer the four research questions defined in Section 1 (RQ1–RQ4), we use the LLM response logs and the Predicted CSAT-Math data described in Section 4.4. By systematically varying input conditions, problem characteristics, and reasoning configurations, we examine LLM mathematical problem-solving ability from multiple angles.

- **RQ1**: Compare overall performance of recent LLMs and derive a ranking based on total score and accuracy over all 46 problems.
- **RQ2**: Analyze which concepts and item types exhibit concentrated errors, considering subject area (Math I/II, Calculus, Geometry, Probability and Statistics), score weight (2/3/4 points), problem format (multiple-choice vs. short-answer), and structural/difficulty factors.
- **RQ3**: Compare how input modality and prompting language affect accuracy, error patterns, and latency.
- **RQ4**: Quantitatively evaluate how changes in reasoning strength affect performance, cost, and latency.

This evaluation framework cross-combines multiple axes—(i) model family and size, (ii) prompting language, (iii) input modality, and (iv) reasoning configuration—to measure LLM math capabilities in a fine-grained manner and to isolate the experimental conditions relevant to each RQ.

Table 4: Summary of experimental settings. We evaluate 24 models across language and input-modality conditions, and conduct additional Reasoning configuration experiments (Reasoning_Effort × Text_Verbosity) for GPT-5.

Axis	# Levels	Levels
Model	24	LLM and multimodal models (Table 2)
Language	2	Korean, English
Input modality	3	Text, Image, Text+Figure
GPT-5 Reasoning_Effort	4	Minimal, Low, Medium, High
GPT-5 Text_Verbosity	3	Low, Medium, High
Base combinations	$24 \times 2 \times 3$	Conditions shared by all models
GPT-5 extra combos	4×3	Effort–Verbosity combinations

4.6 Overall Experimental Configuration

Table 4 summarizes our experimental design. We evaluate 24 models under all combinations of **prompt language (2) × input modality (3)**, and additionally run a reasoning ablation for the GPT-5 family over all `Reasoning_Effort` and `Text_Verbosity` combinations (4 × 3 = 12). This multi-faceted setup enables a precise analysis of math performance, modality sensitivity, language effects, and the trade-off between reasoning strength, cost, and latency.

4.7 Evaluation Metrics

We compute evaluation metrics over the full Korean-CSAT-Math benchmark under a unified scoring scheme. Although human test-takers answer only one elective subject—solving 30 problems in total (22 common + 8 elective)—we require LLMs to answer all three elective sets, yielding 46 problems overall.

- **Score**: Sum of the point values of correctly answered problems. For example, if a model correctly answers five 2-point problems, fourteen 3-point problems, and seventeen 4-point problems, the score is $(2 \times 5) + (3 \times 17) + (4 \times 18) = 133$.
- **Normalized Score**: Total score normalized to a 100-point scale, where 152 is the maximum (Table 1: 74 points for common + 26 each for Probability, Calculus, and Geometry). The formula is $\frac{\text{score} \times 100}{152}$. In the above example, $\frac{133 \times 100}{152} = 87.5$.
- **Latency**: Time taken for the model to generate a response, measured from API invocation until the full response is received. We compute this using `elapsed_sec` in the Predicted CSAT-Math logs and report it in seconds (with millisecond precision).
- **Input Token Usage**: Number of tokens consumed by the prompt.
- **Output Token Usage**: Number of tokens generated in the completion.
- **Total Token Usage**: Sum of input and output tokens (the `total_tokens` field), representing total token consumption per problem.
- **Cost ($)**: Per-problem inference cost, computed from OpenRouter's pricing policy for each model (Table 2). Costs are calculated based on both input and output token usage. Note that even open-weight models (e.g., gpt-oss-20B) incur API costs when accessed via OpenRouter, although they may be freely usable outside this setting. We apply a consistent OpenRouter-based formula to all models.

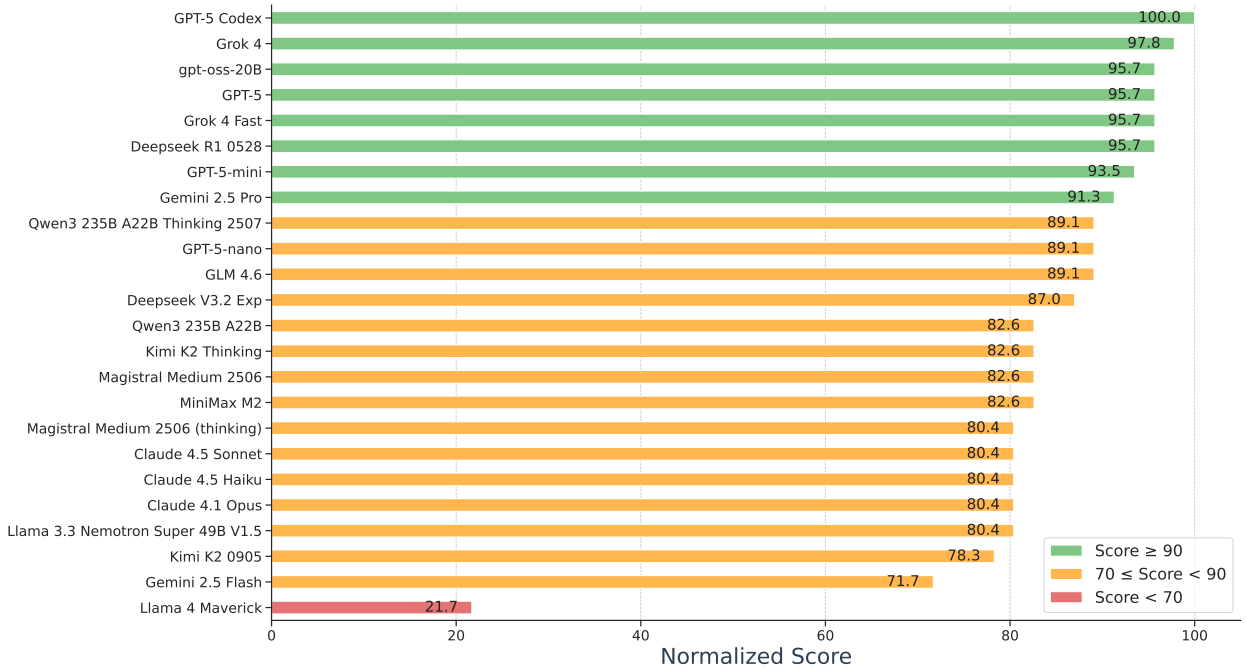


Figure 7: Model-wise performance comparison (input modality: Text-only, prompt language: Korean)

5 LLM Experiment Results

5.1 LLM Score Leaderboard (Text-only / Korean)

The overall performance of each model under the proposed Korean-CSAT LLM evaluation framework is summarized in Table 6 in Appendix A. More detailed results and the latest leaderboard are available on the official website[14]. Below, we present key findings observed under this specific configuration.

The main analysis in this subsection focuses on the condition **input modality: Text-only, prompt language: Korean**. This choice reflects the presence of models that do not support image input and the fact that the target benchmark consists of Korean CSAT problems.

Model-wise performance analysis As shown in Figure 7, **GPT-5 Codex** is the only model that achieved a perfect normalized score of 100 in the main analysis. Other GPT-5 family models (GPT-5: 95.7, GPT-5-mini: 93.5), Grok 4 models (Grok 4: 97.8, Grok 4 Fast: 95.7), and Gemini 2.5 Pro (91.3) also surpassed 90 points, demonstrating strong performance among recent proprietary models. Notably, **gpt-oss-20B achieves performance comparable to the top tier despite being a relatively lightweight open-weight model**.

Most of the remaining models scored in the 70–90 range. For example, Qwen3 235B A22B Thinking 2507 and GLM 4.6 achieved scores close to 90, while DeepSeek V3.2 Exp and Kimi K2 Thinking performed in the 80-point range. In contrast, the Claude family and Gemini 2.5 Flash underperformed relative to other proprietary models, and Llama 4 Maverick ranked last with a score of 21.7. These results indicate substantial variance in mathematical problem-solving capability across LLMs.

Model-wise latency analysis Human test-takers are given 100 minutes to solve 30 problems, whereas our setup requires LLMs to solve all 46 problems. To account for the increased number of questions, we scale the time budget to 152 minutes and evaluate models under both **unlimited-time** and **time-limited** settings.

(1) Unlimited-time performance Figure 8 plots model latency and performance under the Text-only, Korean prompt condition. A notable observation is that most models solved all 46 problems within 100 minutes, i.e., within the human time budget for only 30 problems. Even GPT-5 Codex, the top-performing model, completed all questions in 72 minutes.

[14] `https://isoft.cnu.ac.kr/csat2026/`

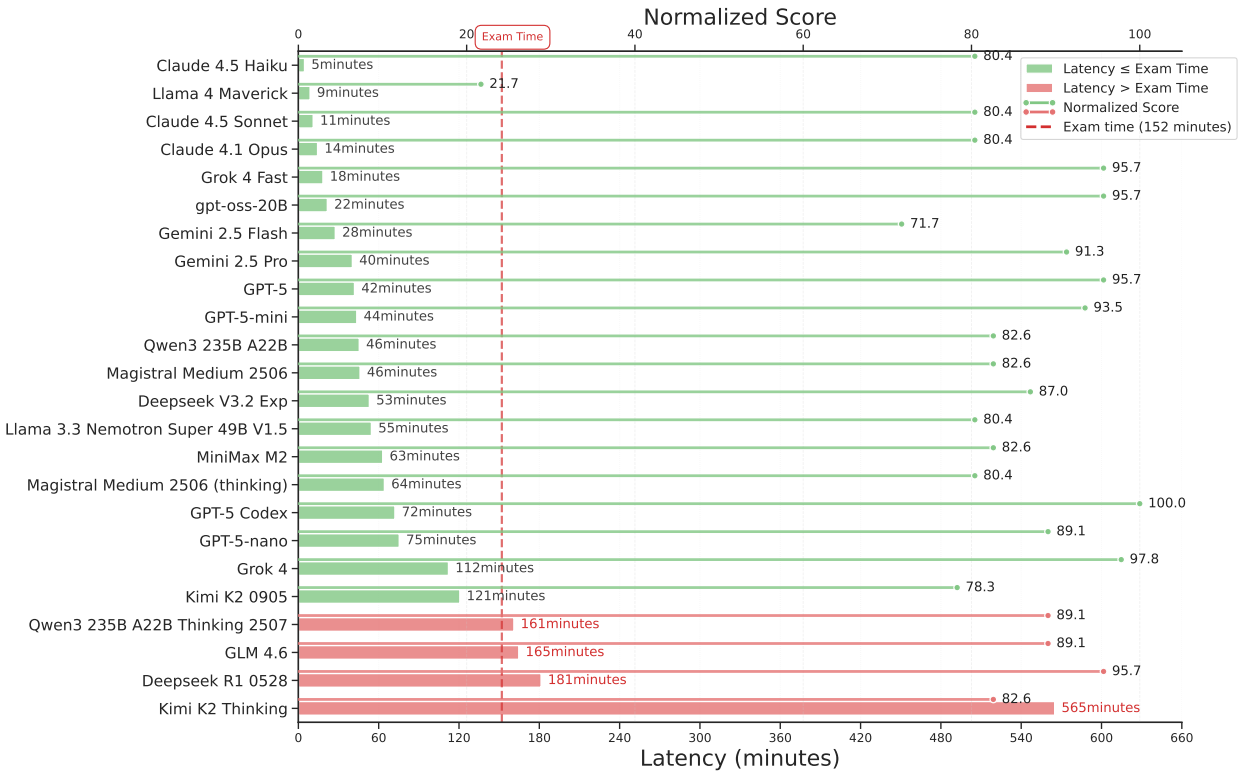


Figure 8: Latency vs. score by model (Text-only, Korean)

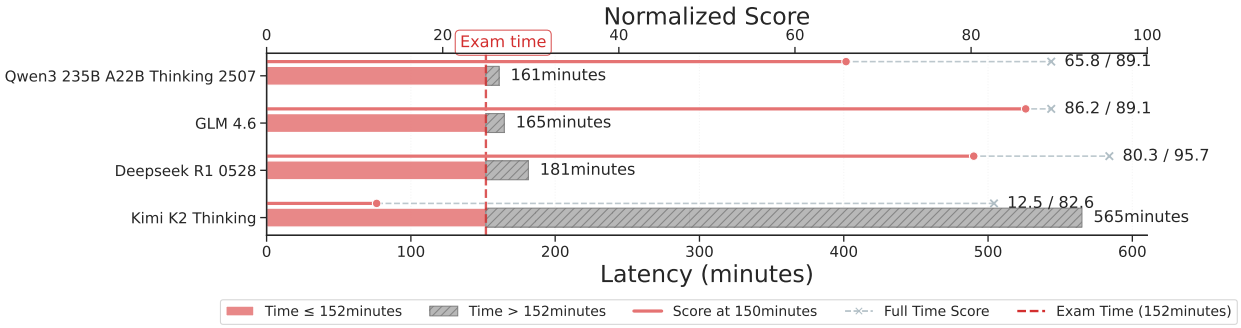


Figure 9: Time-limited performance of models exceeding the 152-minute budget

Table 5: Correlation analysis between latency and score. We report Pearson, Spearman, and Kendall correlations, as well as a simple OLS regression. All correlation coefficients are small and statistically insignificant ($p > 0.05$).

Method	**Correlation**	p**-value**	**N**
Pearson	0.123	5.68×10^{-1}	24
Spearman	0.291	1.68×10^{-1}	24
Kendall τ	0.186	2.19×10^{-1}	24
Log-latency Pearson	0.377	6.98×10^{-2}	24
OLS slope (β_1)	0.016	5.68×10^{-1}	24
OLS R^2	0.015	–	24

Figure 8 shows the latency–score distribution under unlimited reasoning, Text-only input, and Korean prompts.

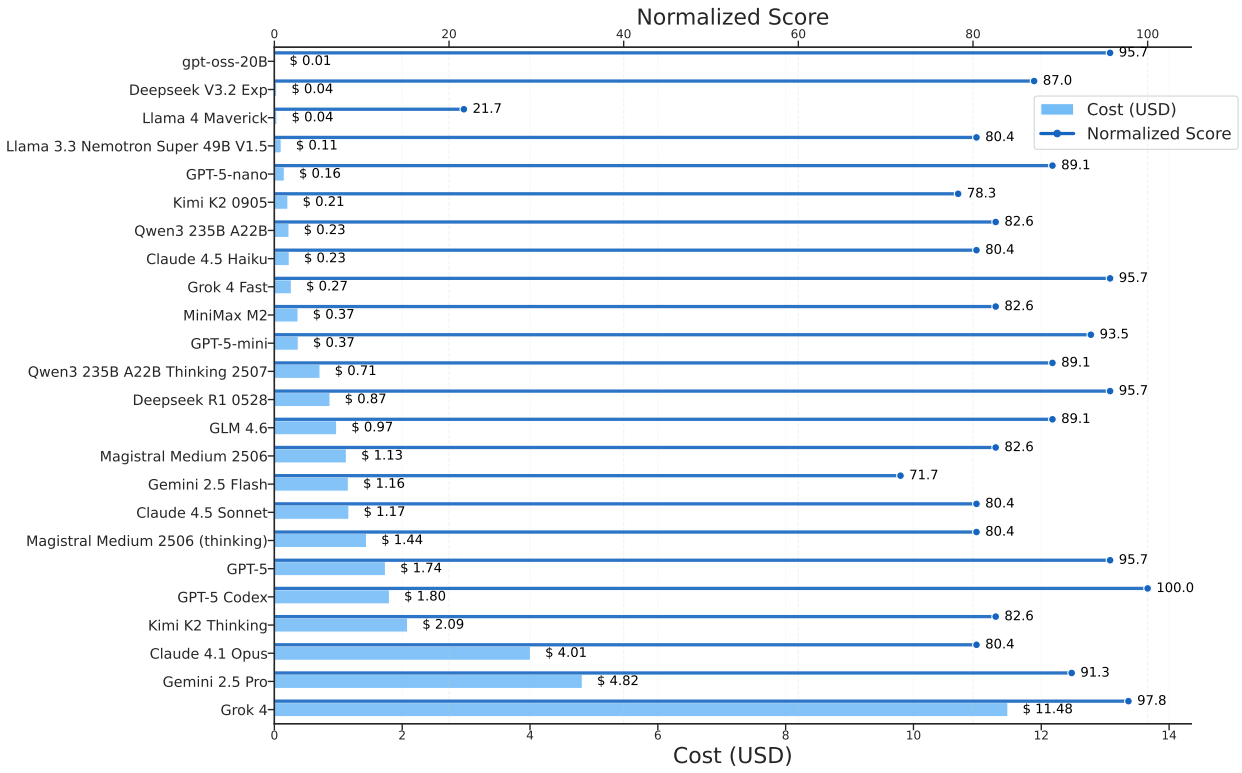


Figure 10: Cost vs. score by model (Text-only, Korean)

We further examine the relationship between latency and score using correlation tests. As summarized in Table 5, Pearson, Spearman, and Kendall coefficients are all small ($|r| < 0.30$) and not statistically significant ($p > 0.05$). In other words, **we do not find evidence of a significant correlation between problem-solving speed and overall model performance**.

(2) Time-limited performance According to Figure 8, four models—Qwen3 235B A22B Thinking 2507, GLM 4.6, DeepSeek R1 0528, and Kimi K2 Thinking—failed to complete all 46 problems within the 152-minute limit. These models are evaluated under both unlimited-time and time-limited settings; Figure 9 compares their scores under the two conditions.

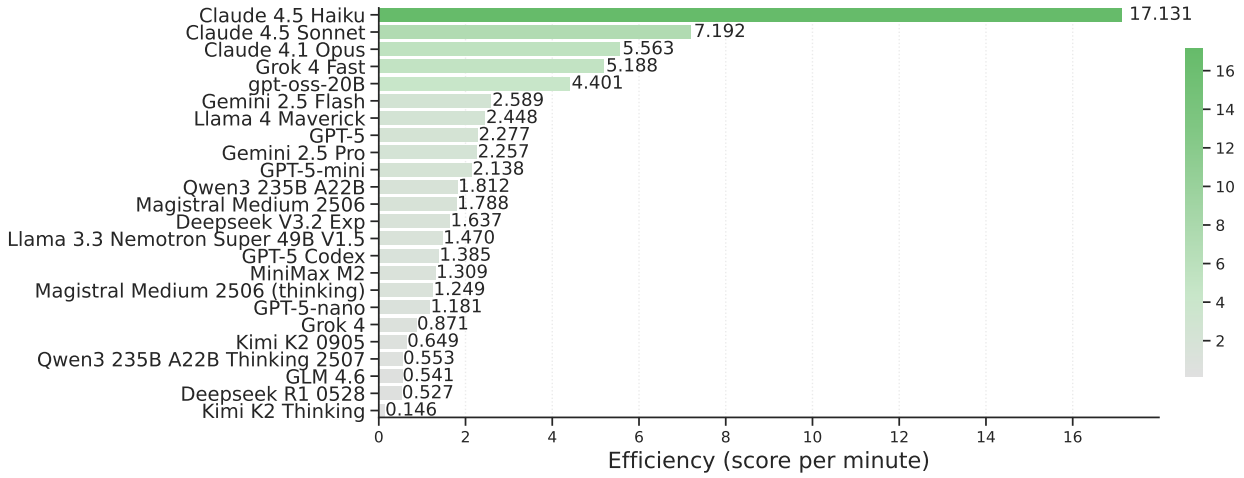
When we only count answers produced within the 152-minute budget, all four models show reduced scores. Kimi K2 Thinking is most affected: its total solving time is 565 minutes (nearly 10 hours), and its score drops from 82.6 to 12.5 when late answers are excluded. Qwen3 235B A22B Thinking 2507 loses about 24 points when the last 9 minutes are cut, and GLM 4.6 loses about 3 points when we exclude its last 13 minutes of responses.

Model-wise cost analysis Figure 10 shows the total API cost for each model to solve all 46 problems. Proprietary models incur relatively high costs: Grok 4 costs $11.48, Gemini 2.5 Pro costs $4.82, and Claude 4.1 Opus costs $4.01. By contrast, open-weight models are much cheaper when accessed via OpenRouter: gpt-oss-20B costs $0.01, DeepSeek V3.2 Exp and Llama 4 Maverick each cost $0.04. Remarkably, gpt-oss-20B achieves a normalized score of 95.7 (3rd place) at the lowest cost ($0.01), yielding an extremely favorable cost–performance ratio.

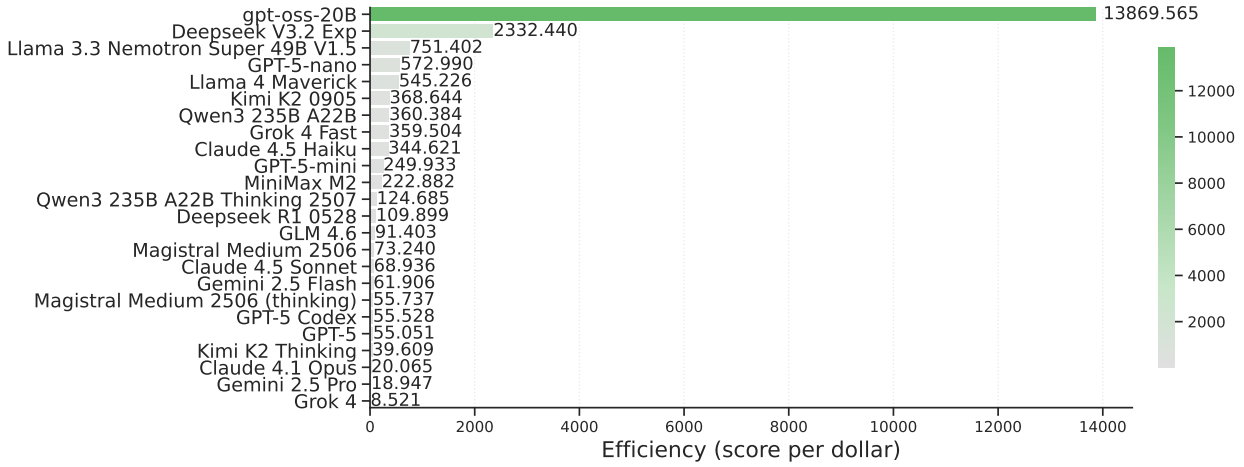
These results show that for some models, cost grows disproportionately relative to performance gains, while others achieve near–state-of-the-art scores at a fraction of the cost.

Model-wise efficiency analysis We define three efficiency scores to jointly analyze performance against time and cost:

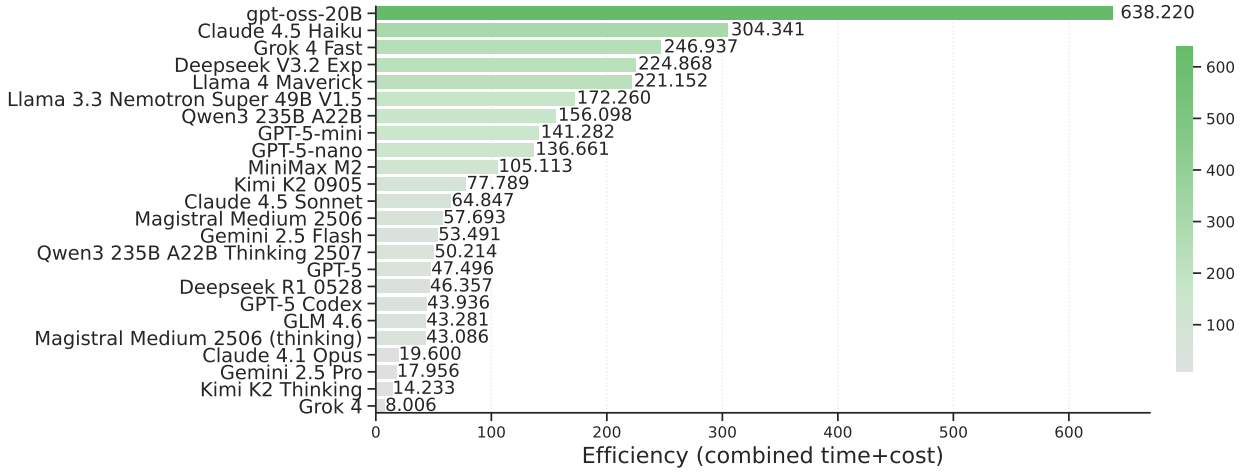
$$(1)\ \mathrm{Eff}_t = \frac{\text{score}}{\text{latency (min)}} \qquad (2)\quad \mathrm{Eff}_c = \frac{\text{score}}{\text{cost (\$)}} \qquad (3)\ \mathrm{Eff}_{t,c} = \frac{\text{score}}{\frac{\text{latency}}{152\ \text{min}} + \frac{\text{cost}}{1\ \$}}$$



(a) Time-based Efficiency Score (Eff_t) across models (Text-only, Korean)



(b) Cost-based Efficiency Score (Eff_c) across models (Text-only, Korean)



(c) Joint time–cost Efficiency Score ($\text{Eff}_{t,c}$) across models (Text-only, Korean)

Figure 11: Efficiency comparison across performance, time, and cost (Text-only, Korean)

Here, Eff_t in (1) captures *time efficiency*: higher values indicate that a model obtains a higher score in less time. Eff_c in (2) captures *cost efficiency*: given the same performance, models with lower cost have higher Eff_c. Finally, $\text{Eff}_{t,c}$ in

(3) is a *joint time–cost efficiency* measure. A high $\text{Eff}_{t,c}$ value indicates that a model achieves high performance while consuming relatively little time and monetary cost.

Figure 11a visualizes **time-normalized performance** using Eff_t. Claude 4.5 Haiku achieves the highest Eff_t, solving all 46 problems in 5 minutes with a score of 80. Other Claude variants rank second and third, followed by Grok 4 Fast and gpt-oss-20B, indicating strong time efficiency.

Figure 11b shows **cost-normalized performance** via Eff_c. gpt-oss-20B achieves a normalized score of 95.7 (3rd) at a cost of $0.01 (1st), resulting in by far the highest Eff_c. DeepSeek V3.2 Exp ranks second, while Grok 4, the most expensive model ($11.48), ranks last on this metric.

Figure 11c visualizes $\text{Eff}_{t,c}$, capturing **overall performance per combined time and monetary resource**. Under this joint metric, gpt-oss-20B is the best model. Claude 4.5 Haiku ranks second: despite its relatively lower cost efficiency, its exceptional time efficiency boosts its overall $\text{Eff}_{t,c}$.

Full results for all input combinations are reported in Appendix B.

In summary, models that are optimal in terms of time efficiency differ from those that are optimal in terms of cost efficiency. **If users prioritize fast responses, or if they need multiple repeated runs, Claude 4.5 Haiku and Claude 4.5 Sonnet are attractive choices; if they prioritize low cost, gpt-oss-20B and DeepSeek V3.2 Exp offer highly favorable performance–resource trade-offs.**

5.2 LLM Performance Under Varying Input and Prompt Conditions

5.2.1 Performance Differences by Model Parameter Size

Performance by Model Size Figure 12 illustrates how model parameter size relates to performance and per-problem latency. As shown in Figure 12a, larger models generally achieve higher scores, but this trend is far from strictly linear. For example, gpt-oss-20B contains roughly seven times fewer active parameters than Qwen3 235B A22B, yet it outperforms the larger model. This demonstrates that parameter count alone is not a definitive determinant of math problem-solving ability.

Relationship Between Model Size and Latency Figure 12b shows that some large models maintain relatively short latencies despite their size, whereas several mid-sized models show extremely long per-problem processing times. This indicates that parameter size produces a complex trade-off structure involving accuracy, latency, and efficiency; no single dimension is sufficient for fully characterizing model performance.

Full results for all modality–language combinations are presented in Appendix C.

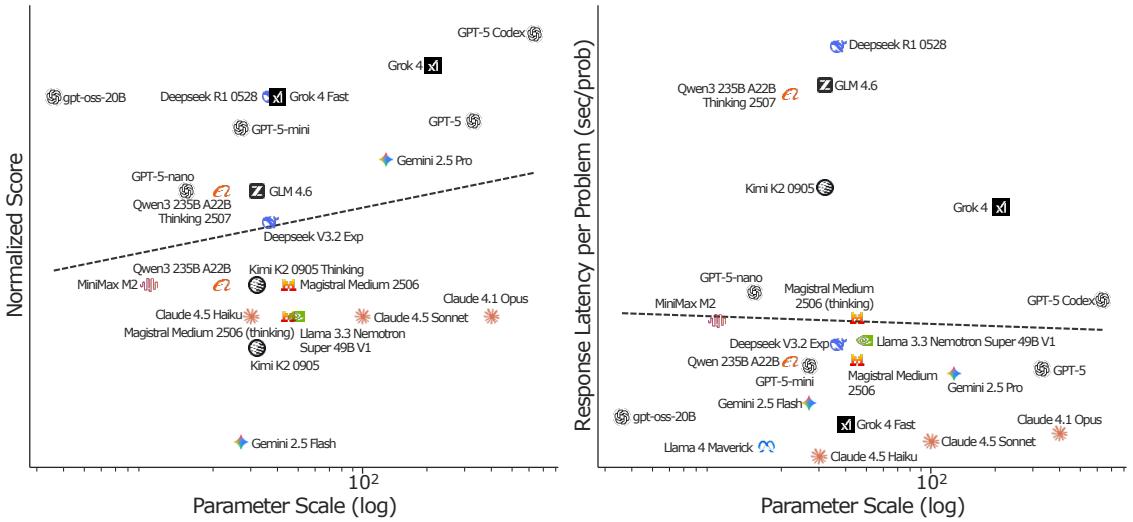
In summary, although larger models generally show stronger performance, **model size is not an absolute predictor of accuracy, and certain smaller models (e.g., gpt-oss-20B) outperform larger models in both accuracy and efficiency.**

5.2.2 Performance Differences by Input Modality

Figure 13a visualizes how input modality affects proprietary models. Key observations are as follows:

- **GPT-5 family**: Achieves the highest scores under Text-only or Text+Figure. GPT-5 Codex reaches **100 points** with Text+Figure. Image-only input lowers performance.
- **Claude family**: Image-only input sharply degrades performance across all variants; Text and Text+Figure perform similarly.
- **Grok family**: Similar to Claude—Image-only performs worst; Text and Text+Figure perform better.
- **Gemini family**: Displays the opposite trend, with **Image-only outperforming other modalities**, indicating strong visual-processing capability.

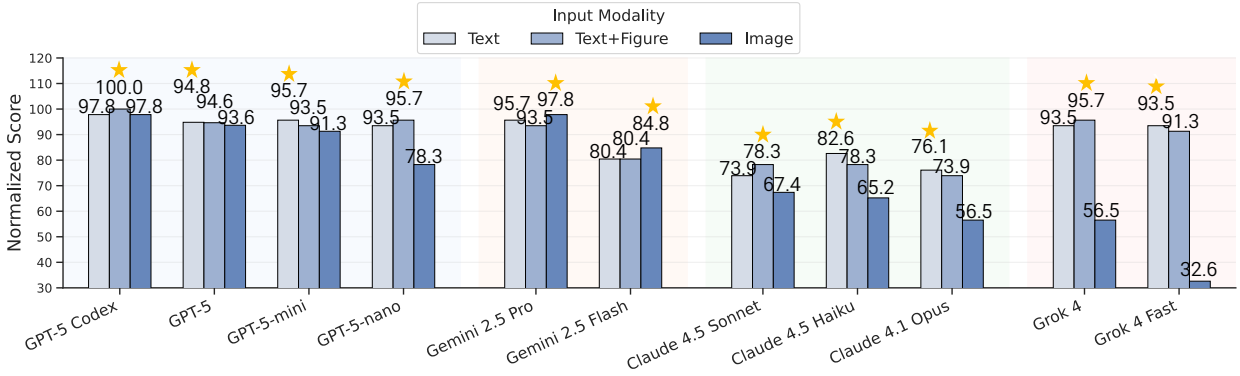
Figure 13b shows win rates by input modality, separated by prompt language and weight accessibility. As in Figure 13a, proprietary models prefer Text and Text+Figure, with lower win rates for Image-only. Open-weight models show slightly different behavior: with Korean prompts, Image input yields a near-zero win rate, but this is based on only three image-capable open-weight models (Qwen3 235B A22B, Qwen3 235B A22B Thinking 2507, Llama 4 Maverick), limiting interpretability.



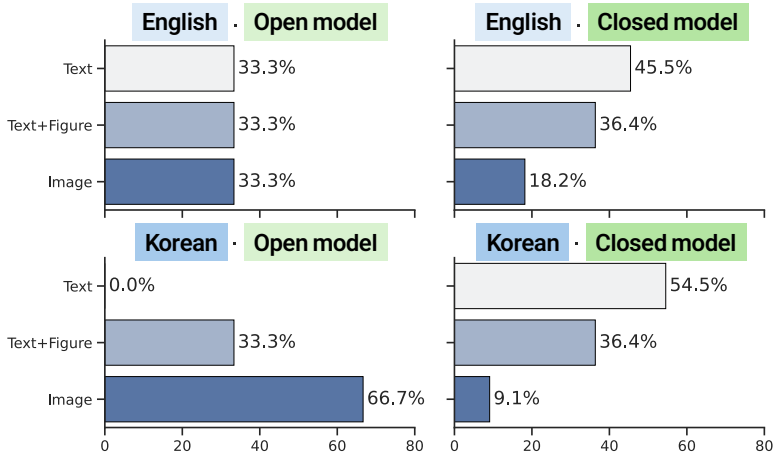
(a) Performance by model size. Llama 4 Maverick (Normalized Score: 21.7) is omitted for legibility.

(b) Latency by model size. Kimi K2 Thinking (737 sec/problem) is omitted for legibility.

Figure 12: Model performance and per-problem latency by parameter size (Text-only, Korean)

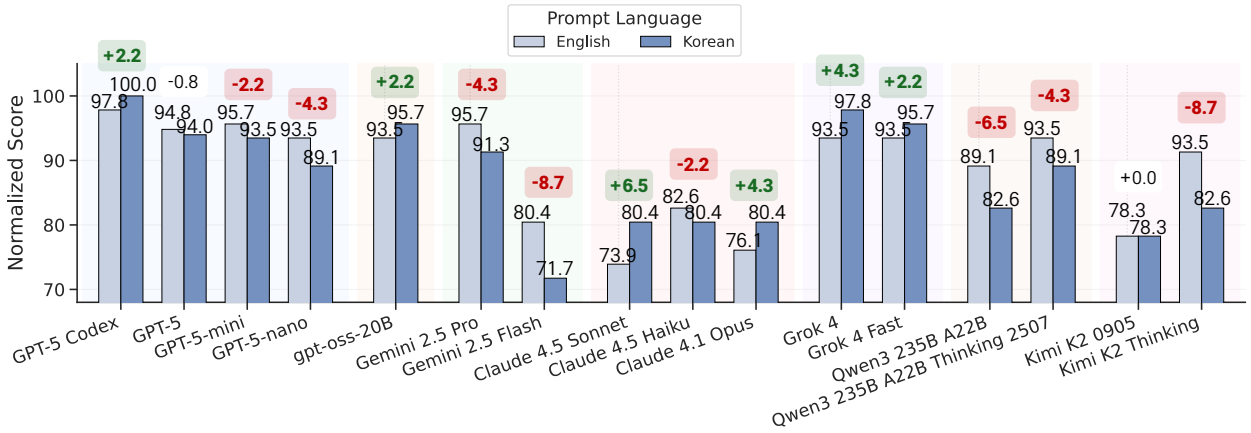


(a) Model-family performance by input modality. Modality with highest score per model is marked with a star.

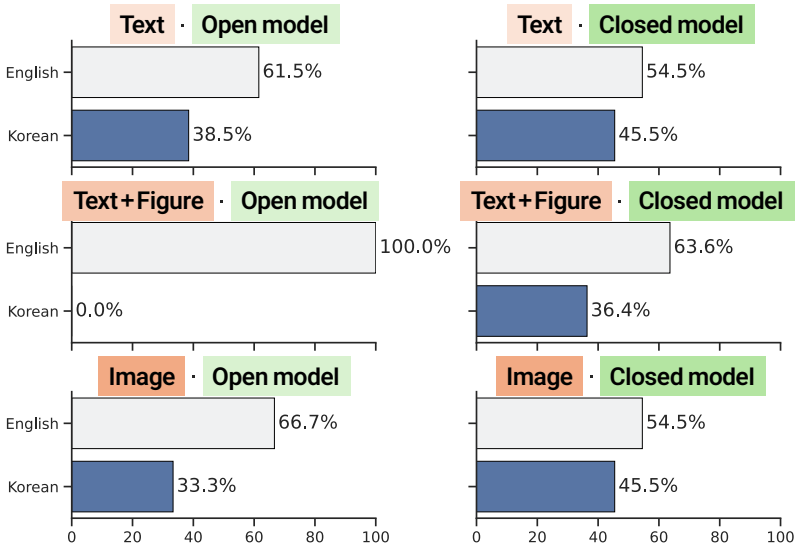


(b) Win rate by input modality, prompt language, and weight accessibility

Figure 13: Performance and win rate comparison across input modalities



(a) Model-family performance by prompting language. Green = Korean better, Red = English better, White = difference < 2 points.



(b) Win rate by prompting language across modalities and accessibility

Figure 14: Performance and win rate comparison across prompting languages

5.2.3 Performance Differences by Prompting Language

Figure 14a shows performance differences between Korean and English prompts across model families. GPT-5 Codex and gpt-oss-20B perform better with Korean prompts, while Claude and Grok models also show an advantage with Korean input. In contrast, Gemini models and major open-weight models (Qwen, Kimi, etc.) exhibit substantially higher performance with English prompts; Korean prompts lead to significant drops in accuracy.

Figure 14b presents win rates by prompting language grouped by input modality and weight accessibility. Across all conditions, **English prompts show a clear advantage**. Even among Text-only inputs, open-weight models benefit more from English prompts than proprietary models. Although results for image-based prompts are influenced by the small number of image-capable open-weight models, English prompts consistently help across modalities.

5.3 Analysis of LLM Performance by Mathematical Problem Characteristics

5.3.1 Performance Differences by Problem Category

Figure 15 presents LLM performance by subject category. The labeled values represent both normalized scores and raw scores (score divided by category maximum).

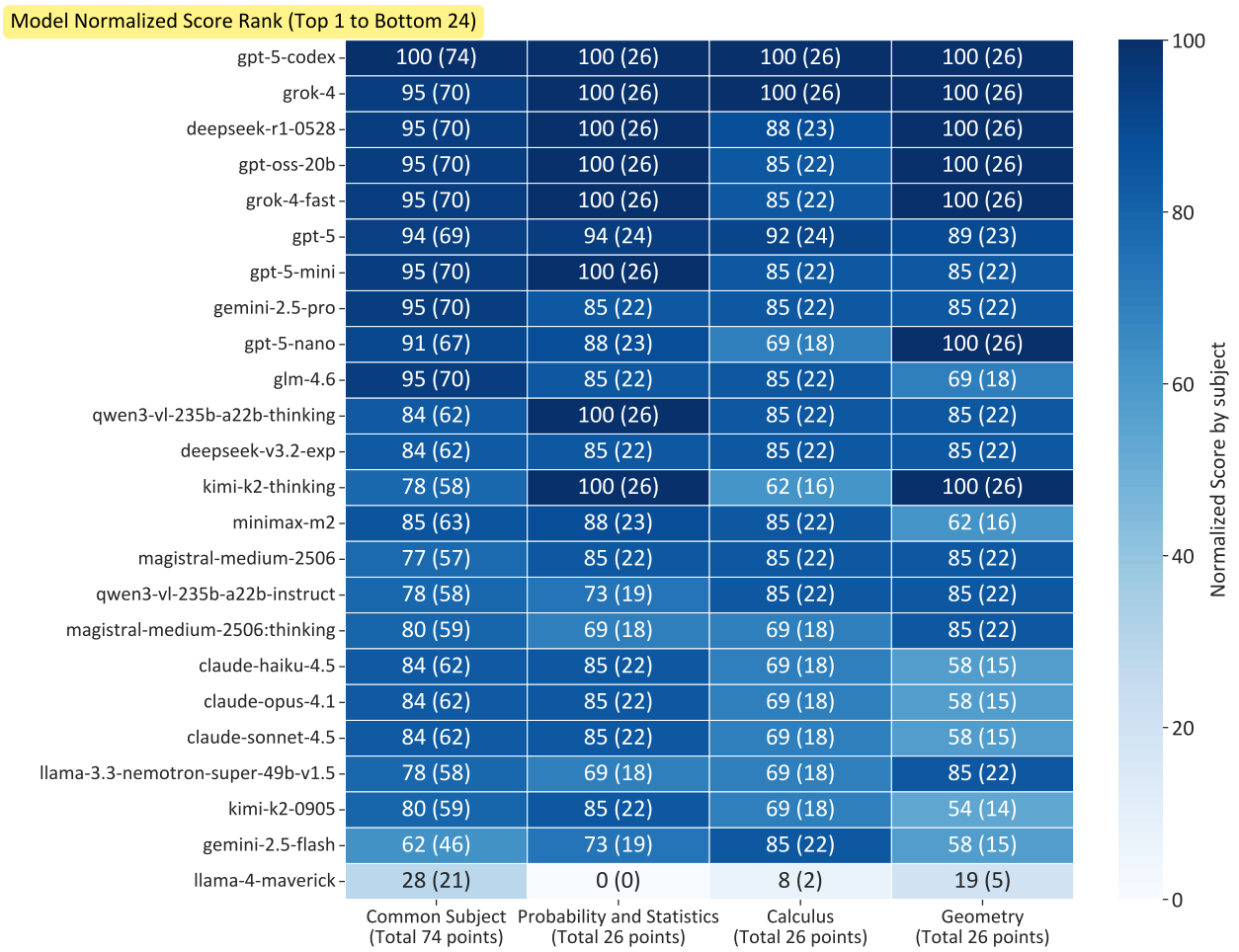


Figure 15: LLM performance by problem category. Normalized scores represent each model's score divided by the maximum for that category. Models are sorted by overall Normalized Score.

Models demonstrate their weakest performance in Calculus, with Geometry ranking as the second most challenging category. The difficulty of Calculus stems from its demand for complex symbolic manipulation and multi-step logical reasoning, whereas Geometry introduces additional variability across models due to its reliance on spatial interpretation and visual understanding.

When averaging across models, performance was lowest in Calculus (77.7pts / 20.2score), followed by Geometry (79.2pts / 20.6score), Common Math (83.9pts / 22.1score), and Probability & Statistics (84.7pts / 22.0score).

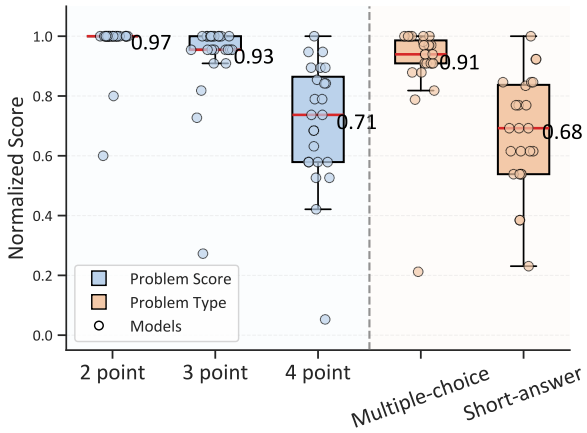
Full results across all modality–language combinations appear in Appendix D.

5.3.2 Performance Differences by Problem Attributes

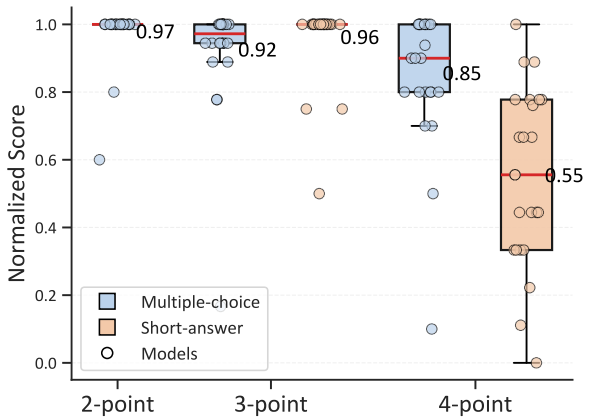
Differences by Problem Type (Multiple-choice vs. Short-answer) Figure 16a compares performance by problem type. LLMs perform substantially better on multiple-choice questions: the average Normalized Score is 91 for five-option problems, compared to 68 for short-answer problems.

However, this difference cannot be attributed solely to format. In CSAT mathematics, the most difficult problems (4-point problems) are overwhelmingly short-answer, meaning problem type and difficulty are deeply intertwined.

Differences by Problem Score (Difficulty Level) A consistent decline in accuracy is observed as problem difficulty increases. Normalized scores for 2- and 3-point questions range from 93 to 97, while 4-point questions drop to 71. This suggests that the human-designed difficulty scale of CSAT mathematics **transfers naturally to LLMs**, making the hardest questions difficult for both humans and models.

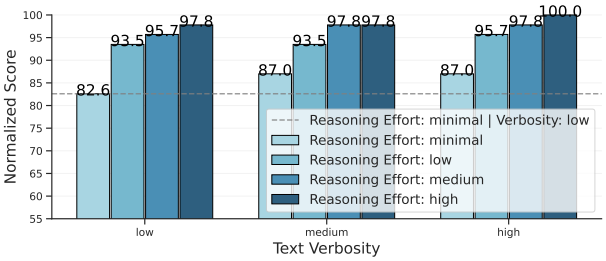


(a) Performance distribution by score (2/3/4 points) and problem type (multiple-choice/short-answer).

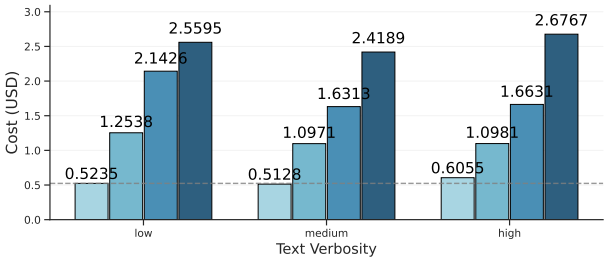


(b) Performance by score–type combinations. (No 2-point multiple-choice problems in 2026 CSAT.)

Figure 16: LLM performance by problem characteristics. (a) examines score and type independently; (b) analyzes joint score–type combinations.



(a) Performance variation across `Reasoning_Effort` and `Text_Verbosity` combinations (GPT-5 family)



(b) Cost variation across `Reasoning_Effort` and `Text_Verbosity` combinations

Figure 17: Comparison of `Reasoning_Effort` and `Text_Verbosity` in GPT-5 configuration

Figure 16b further breaks down performance by the intersection of type (multiple-choice/short-answer) and score (2/3/4). Performance is high for all 2–3 point combinations, but all 4-point combinations—multiple-choice or short-answer—show large accuracy drops and increased variance across models.

Overall, LLMs perform well on easier combinations (2–3 points, multiple-choice), but **struggle significantly on harder combinations, especially 4-point short-answer questions**. Complete results for all modality–language combinations are detailed in Appendix E.

5.4 Effectiveness of Reasoning-Oriented Inference Compared to Baseline LLM Behavior

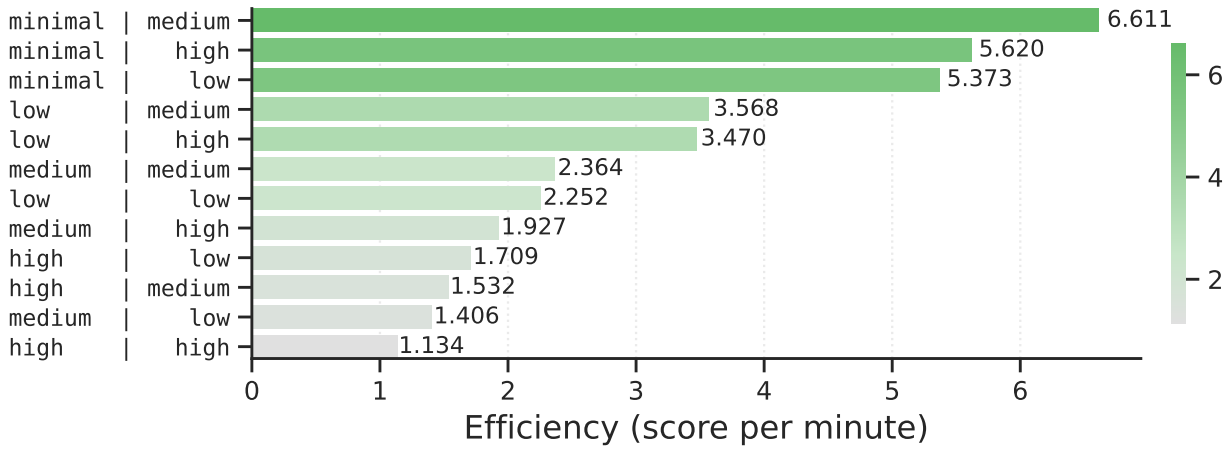
5.4.1 Impact of Reasoning Configuration on Model Performance

Contributions of `Reasoning_Effort` and `Text_Verbosity` Figure 17a shows how different combinations of `Reasoning_Effort` and `Text_Verbosity` affect GPT-5 performance. Increasing `Reasoning_Effort` consistently improves accuracy, whereas increasing `Text_Verbosity` produces almost no performance change.

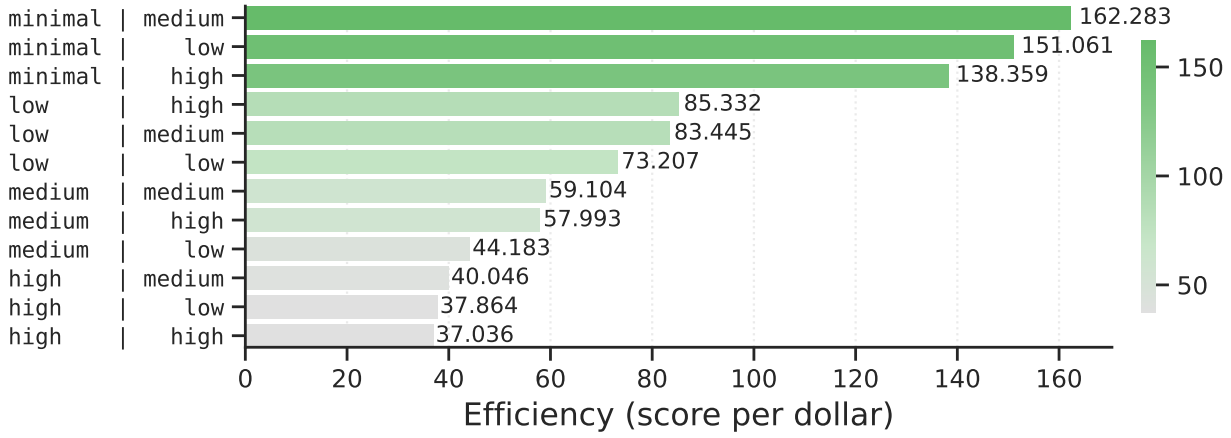
This aligns with the definitions provided in the GPT-5 documentation.[15] `Text_Verbosity` controls the number of output tokens—its effect is limited to explanation length and does not alter internal reasoning. In contrast, `Reasoning_Effort` explicitly controls the number of internal reasoning tokens generated prior to producing an answer, thus directly affecting the model's internal inference process.

Therefore, the observed performance gains primarily reflect increased reasoning effort, not verbosity.

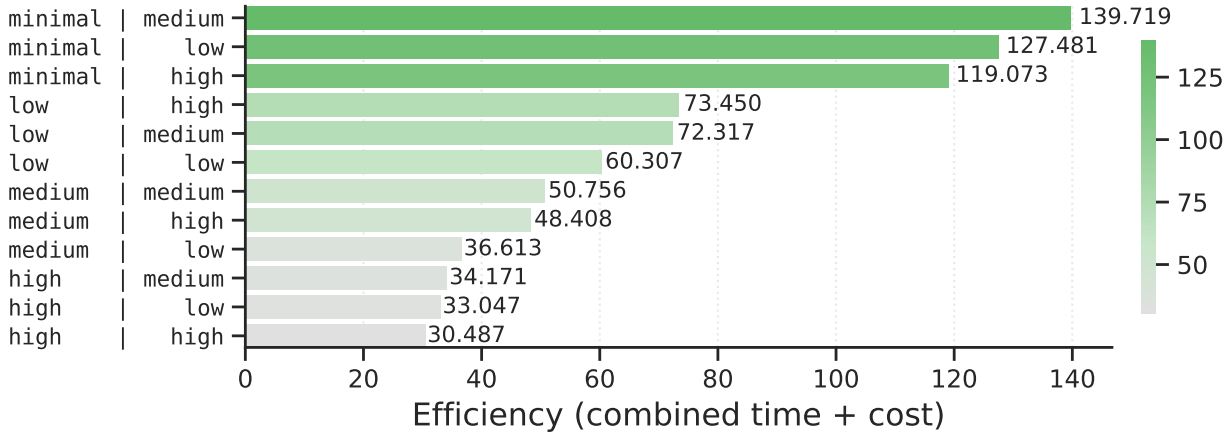
[15] `https://platform.openai.com/docs/guides/latest-model`



(a) Performance–time efficiency under GPT-5 reasoning configurations



(b) Performance–cost efficiency under GPT-5 reasoning configurations



(c) Combined performance–time–cost efficiency under GPT-5 reasoning configurations

Figure 18: Efficiency comparison under GPT-5 reasoning configurations (notation: `Reasoning_Effort | Text_Verbosity`)

5.4.2 Reasoning Cost and the Resulting Efficiency Trade-off

Is extra reasoning cost-effective for GPT-5? As shown in Figure 17, reasoning configurations induce large changes in cost. Thus, evaluating reasoning requires considering not only accuracy but also efficiency. Figures 18a–18c visualize how performance, time, and cost interact across settings.

Overall, increasing `Reasoning_Effort` improves performance but substantially increases both latency and token usage, reducing total efficiency.

The **medium–minimal** combination (`Reasoning_Effort`=medium, `Text_Verbosity`=minimal) achieves the highest values across all Efficiency Scores (performance–time, performance–cost, and performance–time–cost). Minimal verbosity minimizes token consumption while preserving accuracy, giving the best return per unit time and cost.

Conversely, the **high–high** configuration performs worst across all efficiency metrics. while achieving the highest accuracy (100.0) among all effort–verbosity pairs—about 2–3 points higher than typical medium-effort configurations. Although accuracy increases slightly at maximum effort, token usage and cost rise by roughly **4–5 times**, and latency increases significantly. As a result, efficiency scores drop to less than half of the baseline.

Summary: Performance Gains vs. Cost Explosion Increasing reasoning effort in GPT-5 yields only modest performance gains but incurs substantial increases in both cost and latency. In many cases, **performance improvements of only 2–5 points require 4–5 times more tokens**. Thus:

- **High reasoning effort, as expected, increases performance but is not cost-efficient.**
- **Minimal or medium reasoning is more efficient than high reasoning across all metrics.**
- **Baseline GPT-5 or minimal reasoning often provides the best practical trade-off.**
- **High reasoning effort is inefficient for large-scale or budget-limited evaluation settings.**

Overall, **additional reasoning effort benefits performance far less than it increases cost**, but if one aims only to raise the score without regard to cost, it can still be considered. For large-scale or time-sensitive math evaluation tasks, baseline or minimal reasoning remains the most rational choice.

6 Key Findings

This study conducted a comprehensive evaluation of 24 large language models (LLMs) on the 2026 Korean College Scholastic Ability Test (CSAT) Mathematics exam under a strictly zero-leakage setting. Our analysis extends beyond raw accuracy and examines how models differ across input conditions, reasoning configurations, cost profiles, and mathematical content types. The results demonstrate that LLM mathematical reasoning ability must be evaluated from a *multidimensional* perspective—encompassing reasoning strategy, efficiency, modality sensitivity, and domain-specific weaknesses—rather than solely by the number of correct answers.

RQ1. Overall Mathematical Problem-Solving Ability of Current LLMs GPT-5 Codex achieved perfect normalized score (100.0) under multiple input–language configurations, including Text+Figure/English, Text-only/Korean, Image-only/Korean, and Text+Figure/Korean, demonstrating robust performance across both textual and visual modalities. In addition, the GPT-5 base model reached a perfect score under the Text-only/English setting. No other model–condition combination achieved a full score, indicating that perfect CSAT-level mathematical performance is currently limited to the GPT-5 family.

A notable finding is that **gpt-oss-20B**, despite its relatively small parameter count, achieved 95.7 points—surpassing some models more than seven times its size. This supports the observation that model scale is not an absolute determinant of mathematical reasoning ability. When cost and latency are considered jointly, gpt-oss-20B exhibited exceptional efficiency, with Claude 4.5 Haiku, Grok 4 Fast, and DeepSeek V3.2 Exp also offering strong performance-per-cost trade-offs.

Most high-performing models completed all 46 questions within the human exam time equivalent (100 minutes), although a small number of models generated excessively long reasoning traces that led to substantial latency inflation.

RQ2. Effects of Input Conditions on Performance The effect of prompt language varied by model scale. Large, modern models such as Claude 4.5 Sonnet and GPT-5 Codex performed better with Korean prompts, whereas smaller models (e.g., GPT-5-mini, Grok 4 Fast) improved by up to 10 points when given English prompts. This suggests that

smaller models are more sensitive to language imbalance in their training data and exhibit weaker multilingual generalization.

Clear trends also emerged for input modality. **Text-only inputs achieved the highest overall accuracy and outperformed Image-only inputs for nearly all models.** Text+Figure inputs yielded accuracy within 0.3% of Text-only, effectively matching it. Even in geometry questions, text inputs were sufficient, reflecting the exam's design: for the 2026 CSAT, relevant geometric information could be inferred fully from textual descriptions. However, this outcome is specific to this exam format. In settings where visual reasoning is indispensable, modality effects may differ substantially.

RQ3. Effects of Mathematical Problem Characteristics When averaging across all models and condition combinations, Calculus yielded the lowest performance, with an average normalized score of 75.3 (19.6 pts). Geometry followed with 78.4, while Common Mathematics and Probability & Statistics remained in the low-to-mid 80 range. These results indicate that **tasks requiring multi-step logical reasoning (Calculus) and spatial reasoning (Geometry) continue to present the greatest challenges for current LLMs.**

The same pattern was observed across problem attribute difficulty. While 2–3 point problems achieved normalized scores of 0.91–0.97, 4-point problems dropped sharply to 0.68–0.71. This reveals that the **CSAT's human-designed difficulty structure systematically induces performance stratification in LLMs as well**. Problem type showed similar trends: multiple-choice problems (0.93–0.97) outperformed constructed-response problems (0.68–0.71), though this is confounded by difficulty since constructed-response problems disproportionately correspond to high-difficulty questions.

RQ4. Benefits and Limitations of Enhanced Reasoning Within the GPT-5 family, increasing `Reasoning_Effort` from minimal to high raised performance from 82.6 to 100.0 (+17.4), but the token usage rose from approximately 60 tokens/problem to 240–268 tokens/problem, a **four- to five-fold increase**. Efficiency Scores were highest under the minimal-verbosity configurations (1.58–1.70) and dropped sharply under high reasoning effort, reaching as low as 0.37.

The effect of `Text_Verbosity` was negligible, and **performance gains were driven almost entirely by** `Reasoning_Effort`. Notably, minimal or low reasoning sometimes *reduced* performance relative to the baseline, indicating that **forcing additional reasoning onto a model that already performs strong internal inference can be counterproductive**.

These findings confirm that although additional reasoning effort can improve accuracy, it substantially increases computational cost and latency. Achieving an optimal balance of reasoning depth therefore requires **adaptive configuration based on model capability and problem difficulty**. Excessive reasoning is generally inefficient, especially in large-scale or resource-limited evaluation environments.

7 Limitations

This study has several limitations stemming from the experimental design and evaluation environment.

Limited Scope of Model Deployment Settings All model evaluations were conducted through the OpenRouter API to ensure a uniform inference environment, reproducibility, and fair comparison across heterogeneous LLMs. While this approach offers consistency, it also introduces several constraints.

First, LLMs served through OpenRouter may exhibit performance differences compared to their native deployments. Variations in quantization levels, optimization settings, or backend configurations can lead to slight deviations from the models' originally intended performance profiles. Although such discrepancies are not fully avoidable, we mitigated their impact by applying OpenRouter's default configurations uniformly across all models to preserve the fairness of relative comparisons. Future work will complement this setup by running open-weight models locally under tightly controlled closed-examination conditions.

Second, due to OpenRouter access constraints and the selection criteria based on leading models listed in AIME 2025, Korean-specialized LLMs were necessarily excluded from this benchmark. Korean-optimized models may exhibit different sensitivity to prompt language, and their absence limits the generalizability of the multilingual findings. Incorporating such models remains an important direction for further investigation.

Restricted Scope of Reasoning-Enhanced Experiments Experiments involving `Reasoning_Effort` and verbosity adjustments were conducted exclusively on the GPT-5 family. At present, "reasoning," "thinking," and related modes differ significantly in definition and implementation across model providers, making consistent cross-model comparison

difficult. For methodological coherence, we restricted reasoning analysis to the GPT-5 series, which offers formally documented and controllable reasoning parameters. Consequently, reasoning-related conclusions in this study are specific to GPT-5 and should be generalized to other architectures only with caution. Broader cross-model reasoning benchmarks are an important direction for future work.

Absence of Human Benchmark Comparison Although this study evaluates the 2026 CSAT Mathematics exam, official statistics on human test-taker performance were unavailable at the time of analysis and at the time of submission[16]. As a result, we were unable to directly compare LLM performance patterns with those of human examinees. Once the official score distributions and per-problem correctness statistics are released, future work will analyze the gap between "problems difficult for humans" and "problems difficult for LLMs," enabling deeper insight into the cognitive characteristics and limitations of LLMs relative to human reasoning.

8 Conclusion

This study introduces a benchmark and evaluation framework designed to assess the mathematical reasoning capabilities of state-of-the-art LLMs under a fully contamination-free setting using the 2026 Korean CSAT Mathematics exam. All 46 problems were digitized within two hours of the exam's public release, and 24 models were evaluated across systematically varied input modalities, prompt languages, and reasoning intensities.

Experimental results show that while the strongest commercial models achieve scores above 90 on CSAT-level mathematics, their performance varies substantially depending on problem characteristics (e.g., Calculus, high-point problems), input conditions (language and modality), and the presence of time constraints. Although reasoning-enhanced modes improve absolute accuracy, they incur significant computational and monetary costs, making baseline configurations more practical and cost-efficient for many use cases.

This work provides three main contributions. First, it establishes a fully contamination-free evaluation environment that fundamentally eliminates the data leakage concerns present in prior benchmarks. Second, it formalizes a standardized digitization pipeline that converts human-targeted exam materials into LLM-ready evaluation data. Third, it proposes a practical assessment perspective that extends beyond accuracy to incorporate time, cost, and overall efficiency.

The study nevertheless has several limitations, including restrictions arising from the use of OpenRouter APIs, the absence of Korean-specialized models, the reasoning experiments being limited to the GPT-5 family, and the lack of direct comparison with human test-taker statistics. Future work will include longitudinal analyses using CSAT data, alignment with official scoring distributions, extensions to additional subject areas, and evaluation of Korean-optimized models.

Ultimately, this study highlights the need to shift LLM assessment from "how many problems the model gets correct" to "how efficiently and reliably the model problems under real-world constraints." We expect that the proposed evaluation framework will serve as a new reference point for assessing and deploying LLMs in high-stakes exam settings such as the CSAT.

References

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*, 2024.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Konstantin Chernyshev, Vitaliy Polshkov, Ekaterina Artemova, Alex Myasnikov, Vlad Stepanov, Alexei Miasnikov, and Sergei Tilga. U-math: A university-level benchmark for evaluating mathematical skills in llms. *arXiv preprint arXiv:2412.03205*, 2024.

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

[16] 2025.11.27

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8706–8719, 2024.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.98. URL `https://aclanthology.org/2021.emnlp-main.98/`.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Eric Guo, Robert Baseggio, Colin Chen, Dawn Chen, Jesse Choi, Andy Dang, et al. Measuring mathematical problem solving with the MATH dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Naila Shafirni Hidayat, Muhammad Dehan Al Kautsar, Alfan Farizki Wicaksono, and Fajri Koto. Simulating training data leakage in multiple-choice benchmarks for llm evaluation. *arXiv preprint arXiv:2505.24263*, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. GSM-Plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers, 2024.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. *arXiv preprint arXiv:2405.12209*, 2024.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

Yujun Mao, Yoon Kim, and Yilun Zhou. CHAMP: A competition-level dataset for fine-grained analyses of LLMs' mathematical reasoning capabilities. *arXiv preprint arXiv:2401.06961*, 2024.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.759. URL `https://aclanthology.org/2022.emnlp-main.759/`.

Cheonbok Park, Jeonghoon Kim, Joosung Lee, Sanghwan Bae, Jaegul Choo, and Kang Min Yoo. Cross-lingual collapse: How language-centric foundation models shape reasoning in large language models. *arXiv preprint arXiv:2506.05850*, 2025.

Jake Snell et al. Scaling LLM test-time compute optimally can be more efficient than scaling model size. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with the MATH-Vision dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*, 2024.

Zhanke Zhou, Zhaocheng Zhu, Xuan Li, Mikhail Galkin, Xiao Feng, Sanmi Koyejo, Jian Tang, and Bo Han. Landscape of thoughts: Visualizing the reasoning process of large language models. *arXiv preprint arXiv:2503.22165*, 2025.

A Full Model Results: Performance, Latency, and Cost

This appendix supplements the summary provided in the main text by presenting the complete results for all models across every combination of input modality and prompt language. Table 6 reports the normalized performance scores for each model–condition pair. Table 7 summarizes the average response latency under the same conditions, and Table 8 provides the total cost incurred to evaluate each model.

Table 6: Normalized performance scores across all models and all input–language combinations.

Model Type	Model	English			Korean		
		Text	Image	Text+Figure	Text	Image	Text+Figure
Closed-weight	GPT-5 Codex	97.8	97.8	100.0	100.0	100.0	100.0
	GPT-5	95.7	100.0	95.7	95.7	97.8	97.8
	GPT-5-mini	95.7	91.3	93.5	93.5	91.3	93.5
	GPT-5-nano	93.5	78.3	95.7	89.1	67.4	89.1
	Grok 4	93.5	56.5	95.7	97.8	52.2	95.7
	Grok 4 Fast	93.5	32.6	91.3	95.7	37.0	97.8
	Claude 4.5 Sonnet	73.9	67.4	78.3	80.4	71.7	82.6
	Claude 4.5 Haiku	82.6	65.2	78.3	80.4	67.4	82.6
	Claude 4.1 Opus	76.1	56.5	73.9	80.4	69.6	76.1
	Gemini 2.5 Pro	95.7	97.8	93.5	91.3	93.5	93.5
	Gemini 2.5 Flash	80.4	84.8	80.4	71.7	73.9	71.7
Open-weight	gpt-oss-20B	93.5	–	–	95.7	–	–
	Qwen3 235B A22B	89.1	73.9	91.3	82.6	69.6	84.8
	Qwen3 235B A22B Thinking 2507	93.5	84.8	91.3	89.1	97.8	84.8
	Deepseek R1 0528	93.5	–	–	95.7	–	–
	Deepseek V3.2 Exp	91.3	–	–	87.0	–	–
	GLM 4.6	87.0	–	–	89.1	–	–
	Magistral Medium 2506	71.7	–	–	82.6	–	–
	Magistral Medium 2506 Thinking	84.8	–	–	80.4	–	–
	MiniMax M2	97.8	–	–	82.6	–	–
	Llama 4 Maverick	23.9	65.2	32.6	21.7	43.5	28.3
	Llama 3.3 Nemotron Super 49B V1.5	80.4	–	–	80.4	–	–
	Kimi K2 0905	78.3	–	–	78.3	–	–
	Kimi K2 Thinking	91.3	–	–	82.6	–	–

Table 7: Average latency (ms) across all models and input–language combinations.

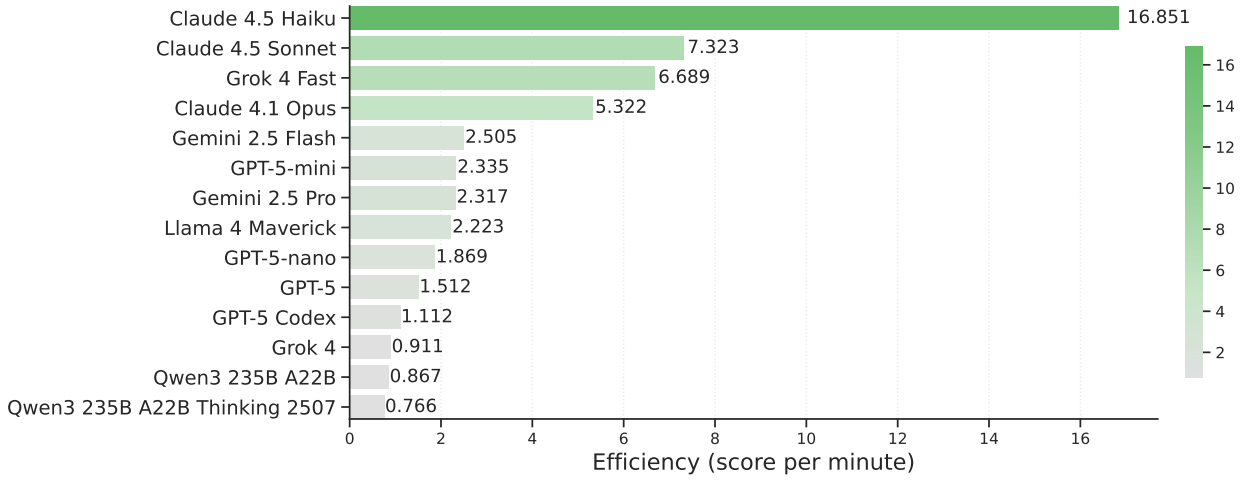
Model Type	Model	English			Korean		
		Text	Image	Text+Figure	Text	Image	Text+Figure
Closed-weight	GPT-5 Codex	5457.4	5808.1	6130.2	4333.3	7426.4	5393.7
	GPT-5	2134.3	4839.8	3561.1	2521.6	4832.4	3882.0
	GPT-5-mini	2297.2	2221.5	2012.6	2623.8	2875.0	2402.5
	GPT-5-nano	3081.7	2816.4	3354.8	4527.9	3454.1	2860.4
	Grok 4	6234.1	16463.0	4371.7	6734.4	16875.1	6302.7
	Grok 4 Fast	840.3	2866.6	861.6	1106.7	2812.5	868.4
	Claude 4.5 Sonnet	641.3	719.3	655.3	670.7	751.7	676.8
	Claude 4.5 Haiku	275.9	308.9	284.4	281.6	343.0	294.1
	Claude 4.1 Opus	780.1	927.9	829.2	867.1	958.6	858.0
	Gemini 2.5 Pro	2742.4	2923.7	2581.8	2427.3	2770.0	2421.2
	Gemini 2.5 Flash	1537.0	1461.4	1408.9	1661.7	1736.1	1717.4
Open-weight	gpt-oss-20B	1729.8	-	-	1304.6	-	-
	Qwen3 235B A22B	2890.0	3888.1	2309.5	2735.6	6997.7	5865.2
	Qwen3 235B A22B Thinking 2507	12616.1	9157.2	9288.6	9665.3	10350.5	6639.4
	Deepseek R1 0528	11540.0	-	-	10885.7	-	-
	Deepseek V3.2 Exp	3161.0	-	-	3057.1	-	-
	GLM 4.6	13152.2	-	-	9884.7	-	-
	Magistral Medium 2506	3986.7	-	-	2771.8	-	-
	Magistral Medium 2506 (thinking)	4998.6	-	-	3862.6	-	-
	MiniMax M2	4037.6	-	-	3786.8	-	-
	Llama 4 Maverick	589.7	1151.0	723.6	531.9	915.0	1186.2
	Llama 3.3 Nemotron Super 49B V1.5	3235.2	-	-	3262.7	-	-
	Kimi K2 0905	10250.8	-	-	7242.8	-	-
	Kimi K2 Thinking	26214.9	-	-	33906.9	-	-

Table 8: Total evaluation cost (USD) across all models and input–language combinations.

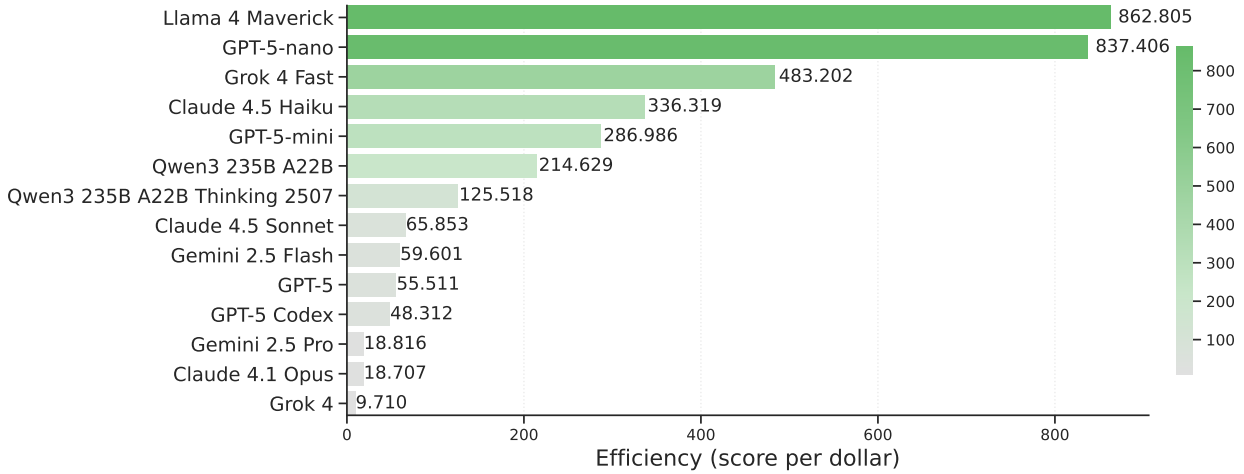
Model Type	Model	English			Korean		
		Text	Image	Text+Figure	Text	Image	Text+Figure
Closed-weight	GPT-5 Codex	2.5181	1.9222	2.0817	1.8009	2.3985	2.0699
	GPT-5	1.4378	1.6743	1.7073	1.7384	1.7896	1.7618
	GPT-5-mini	0.3187	0.3022	0.2683	0.3741	0.3699	0.3258
	GPT-5-nano	0.1063	0.1126	0.1250	0.1555	0.1287	0.1064
	Grok 4	10.5872	24.3791	7.5040	11.4779	26.1858	9.8555
	Grok 4 Fast	0.2028	0.6162	0.2071	0.2662	0.4924	0.2024
	Claude 4.5 Sonnet	1.0253	1.2765	1.1099	1.1663	1.4694	1.2543
	Claude 4.5 Haiku	0.2084	0.2415	0.2214	0.2333	0.2842	0.2456
	Claude 4.1 Opus	3.3942	3.8985	3.6921	4.0069	4.3796	4.0680
	Gemini 2.5 Pro	5.3461	5.8185	5.2356	4.8186	5.4515	4.9693
	Gemini 2.5 Flash	1.0569	1.0381	0.9517	1.1582	1.2454	1.2030
Open-weight	gpt-oss-20B	0.0075	-	-	0.0069	-	-
	Qwen3 235B A22B	0.1886	0.3279	0.1835	0.2292	0.5302	0.3951
	Qwen3 235B A22B Thinking 2507	0.8418	0.8858	0.7319	0.7146	0.6988	0.6756
	Deepseek R1 0528	1.0082	-	-	0.8708	-	-
	Deepseek V3.2 Exp	0.0332	-	-	0.0373	-	-
	GLM 4.6	1.4065	-	-	0.9748	-	-
	Magistral Medium 2506	1.6690	-	-	1.1278	-	-
	Magistral Medium 2506 (thinking)	2.1733	-	-	1.4425	-	-
	MiniMax M2	0.4169	-	-	0.3706	-	-
	Llama 4 Maverick	0.0440	0.0378	0.0353	0.0398	0.0402	0.0328
	Llama 3.3 Nemotron Super 49B V1.5	0.1092	-	-	0.1070	-	-
	Kimi K2 0905	0.2891	-	-	0.2124	-	-
	Kimi K2 Thinking	1.4857	-	-	2.0854	-	-

B Full Results on Model-wise Performance–Time–Cost Efficiency

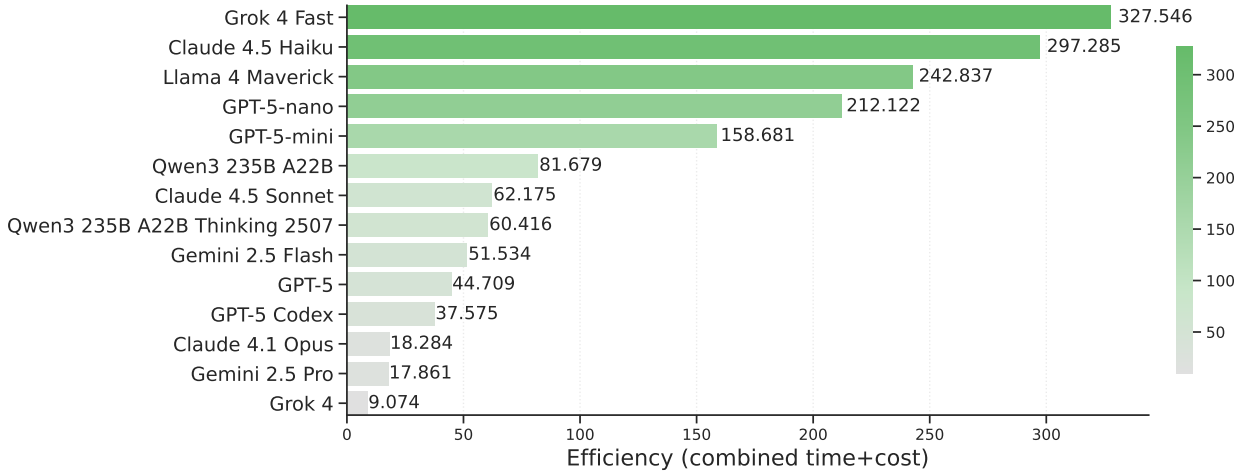
This appendix presents the full set of Efficiency Scores by model, extending the analysis in Section 5.1 to all input–language combinations (Text / Image / Text+Figure $\times$ Korean / English). Figure 19 compares Efficiency Scores for the (Text+Figure, Korean) condition; Figure 20 for the (Image-only, Korean) condition; Figure 21 for the (Text-only, English) condition; Figure 22 for the (Text+Figure, English) condition; and Figure 23 for the (Image-only, English) condition. Models that do not support image input are excluded from the Image-only and Text+Figure conditions. The corresponding plots for the (Text-only, Korean) condition are provided in Figure 11 in the main text.



(a) Comparison of performance–time Efficiency Scores **(input modality: Text+Figure, prompt language: Korean)**

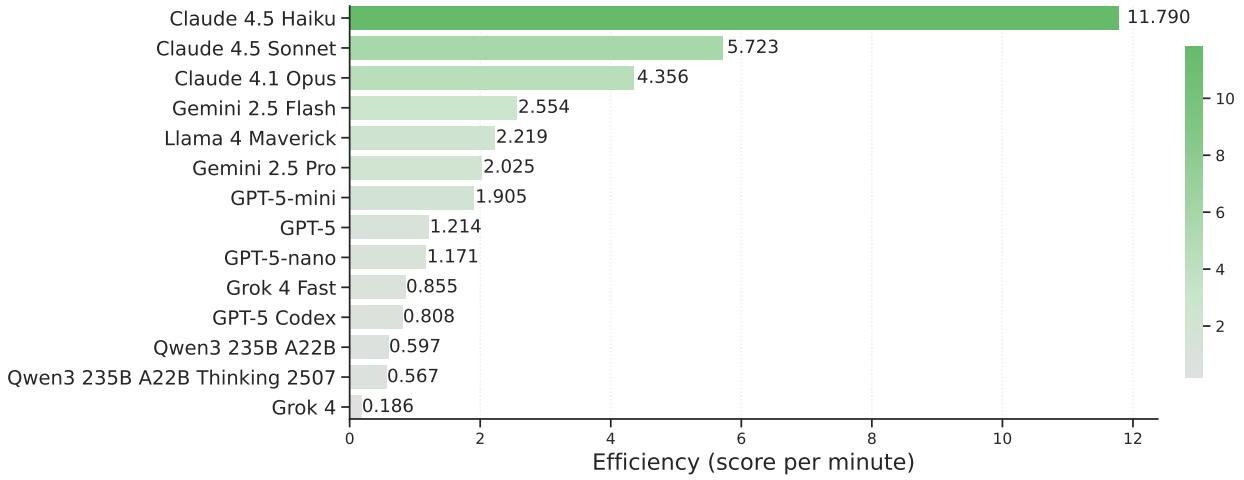


(b) Comparison of performance–cost Efficiency Scores **(input modality: Text+Figure, prompt language: Korean)**

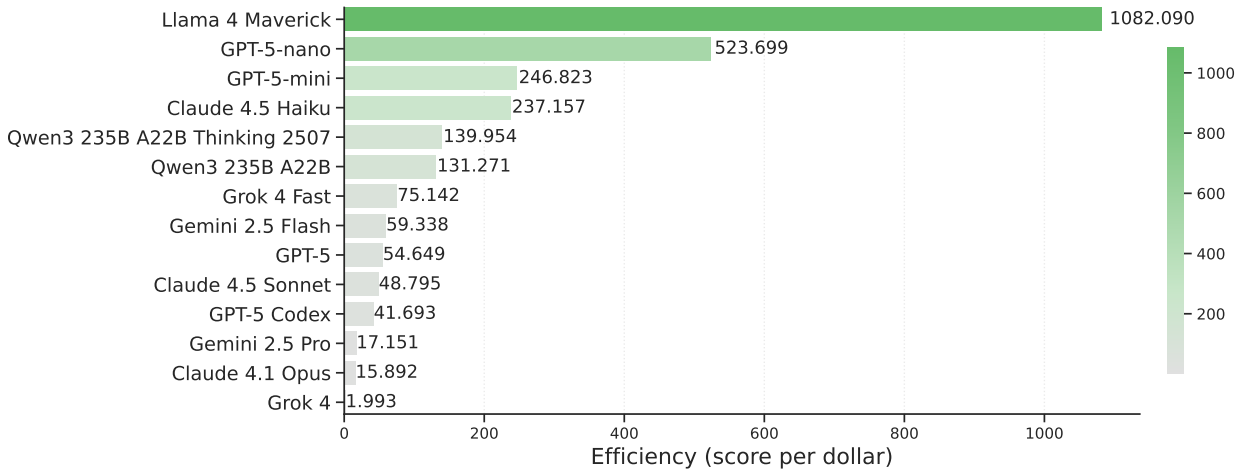


(c) Comparison of performance–time–cost Efficiency Scores **(input modality: Text+Figure, prompt language: Korean)**

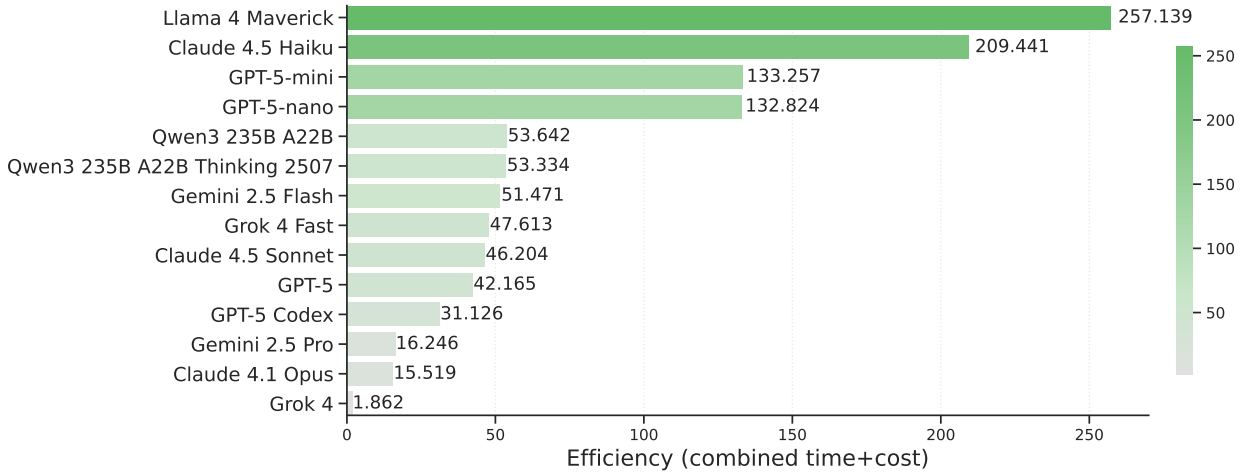
Figure 19: Comparison of performance–time–cost Efficiency Scores across models **(input modality: Text+Figure, prompt language: Korean)**



(a) Comparison of performance–time Efficiency Scores **(input modality: Image-only, prompt language: Korean)**

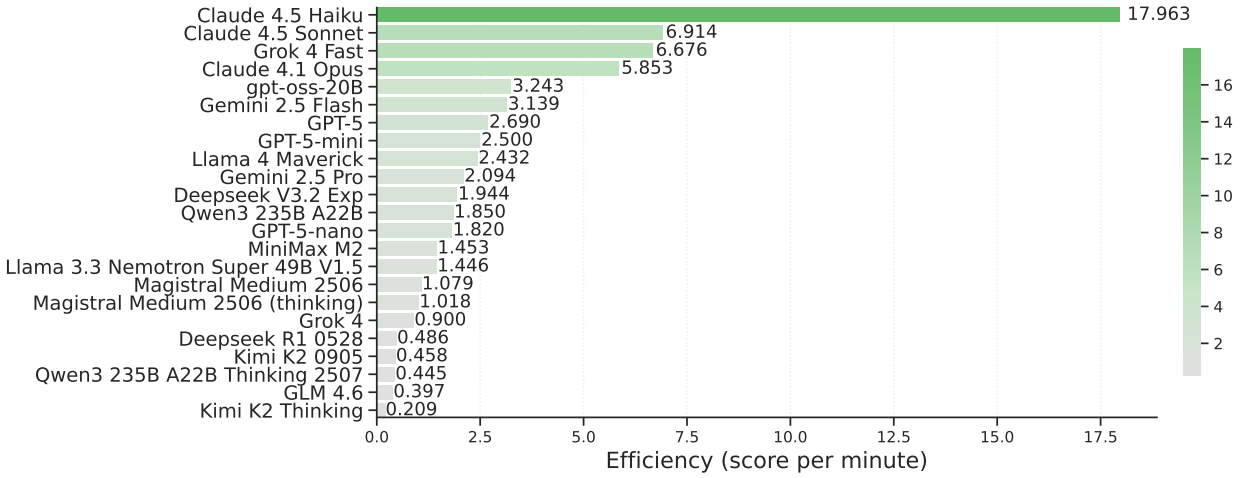


(b) Comparison of performance–cost Efficiency Scores **(input modality: Image-only, prompt language: Korean)**

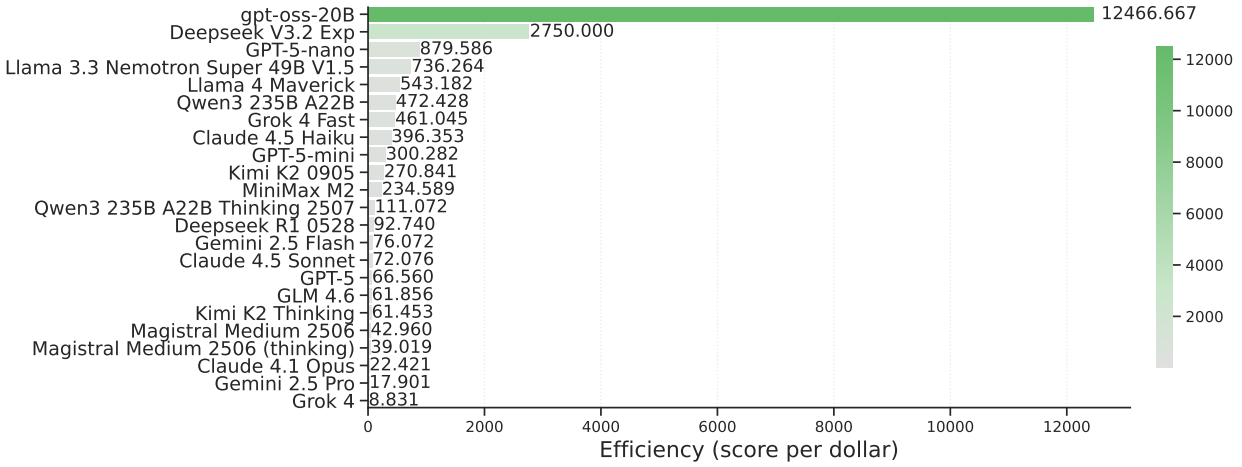


(c) Comparison of performance–time–cost Efficiency Scores **(input modality: Image-only, prompt language: Korean)**

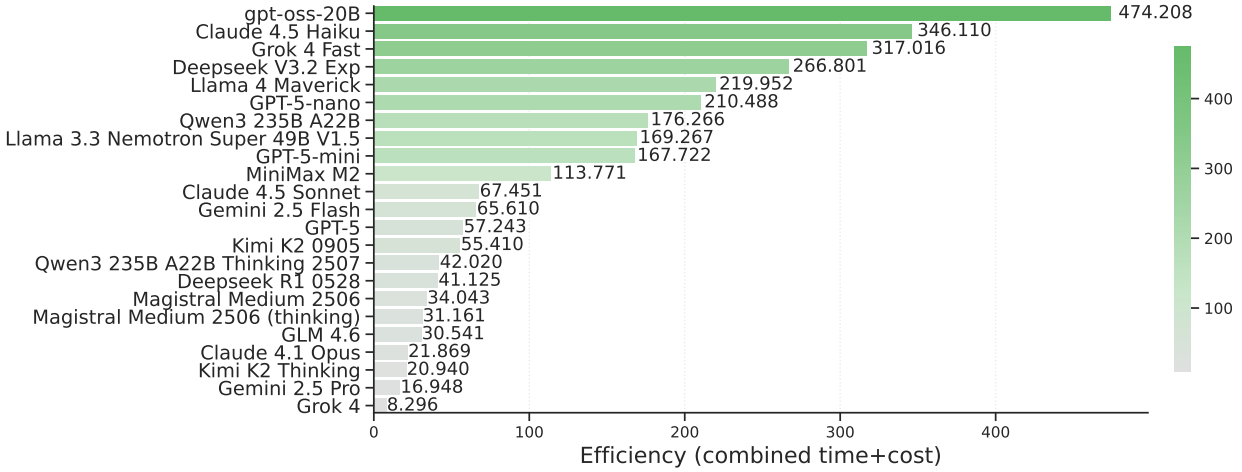
Figure 20: Comparison of performance–time–cost Efficiency Scores across models **(input modality: Image-only, prompt language: Korean)**



(a) Comparison of performance–time Efficiency Scores **(input modality: Text-only, prompt language: English)**

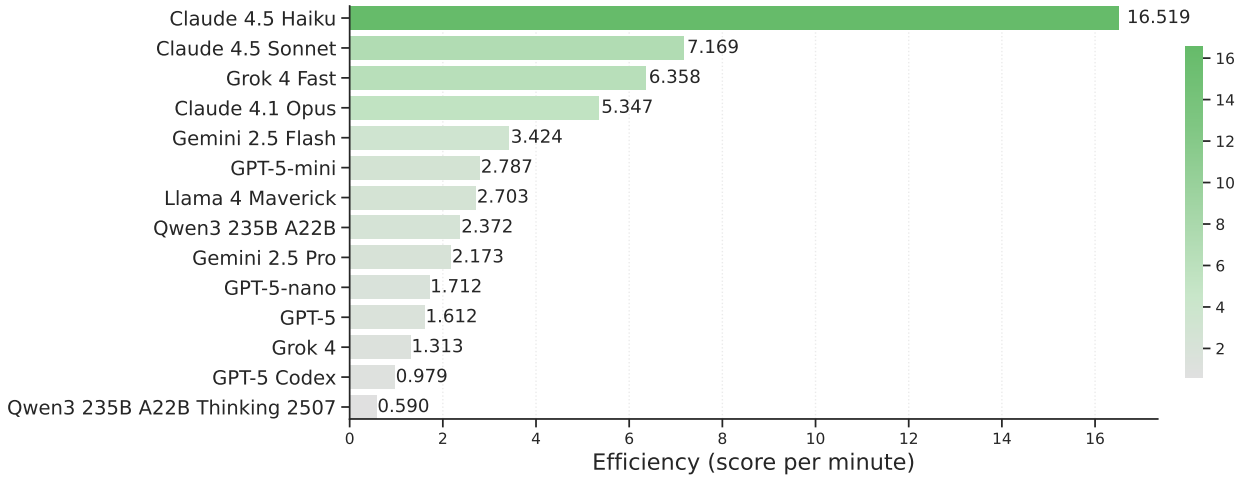


(b) Comparison of performance–cost Efficiency Scores **(input modality: Text-only, prompt language: English)**

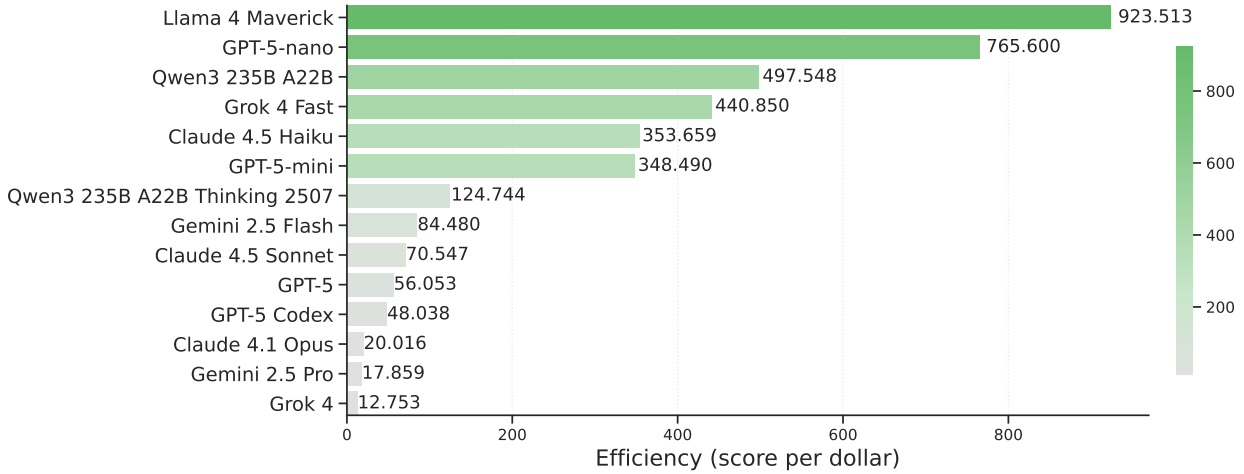


(c) Comparison of performance–time–cost Efficiency Scores **(input modality: Text-only, prompt language: English)**

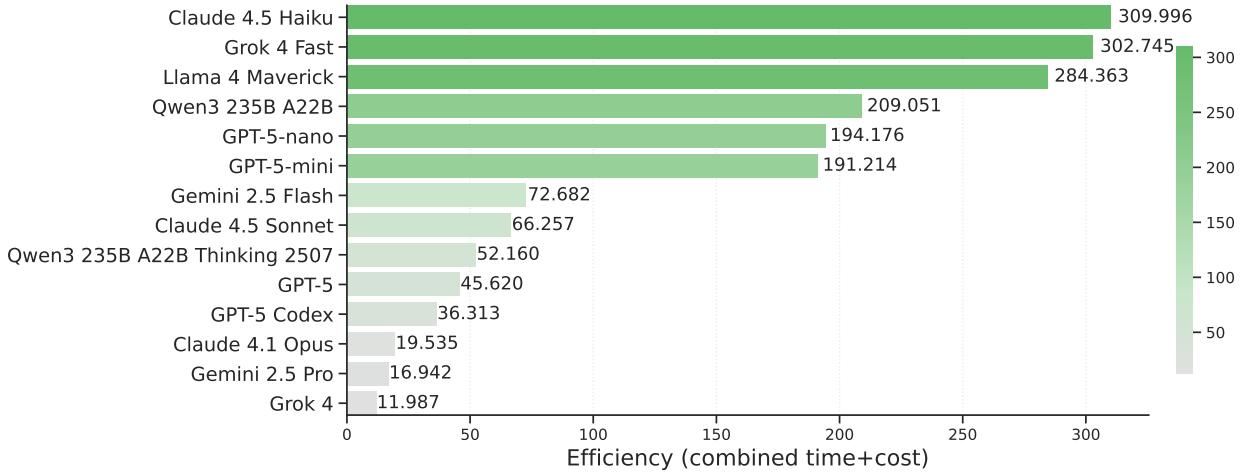
Figure 21: Comparison of performance–time–cost Efficiency Scores across models **(input modality: Text-only, prompt language: English)**



(a) Comparison of performance–time Efficiency Scores **(input modality: Text+Figure, prompt language: English)**

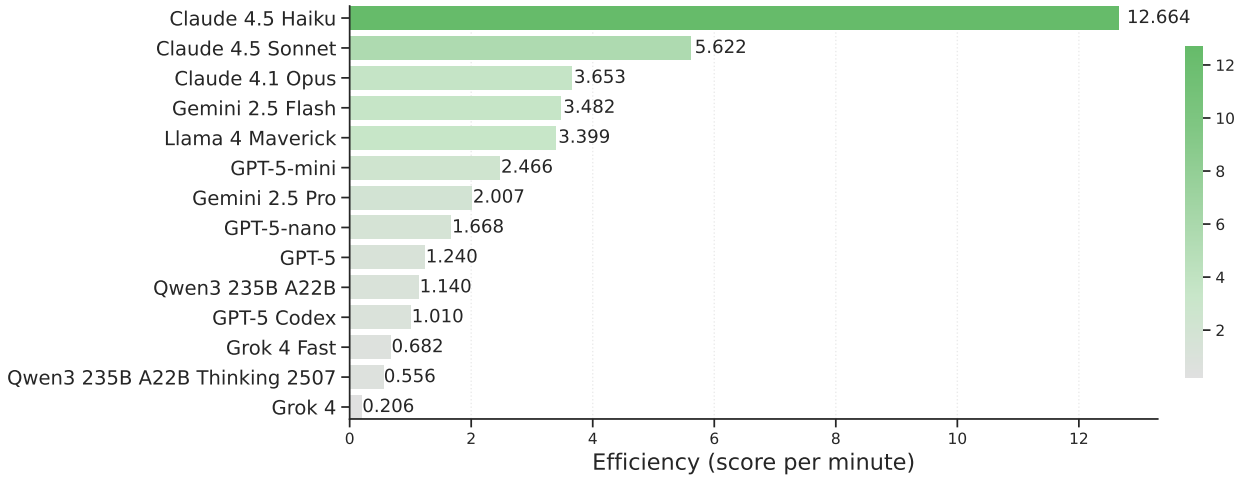


(b) Comparison of performance–cost Efficiency Scores **(input modality: Text+Figure, prompt language: English)**

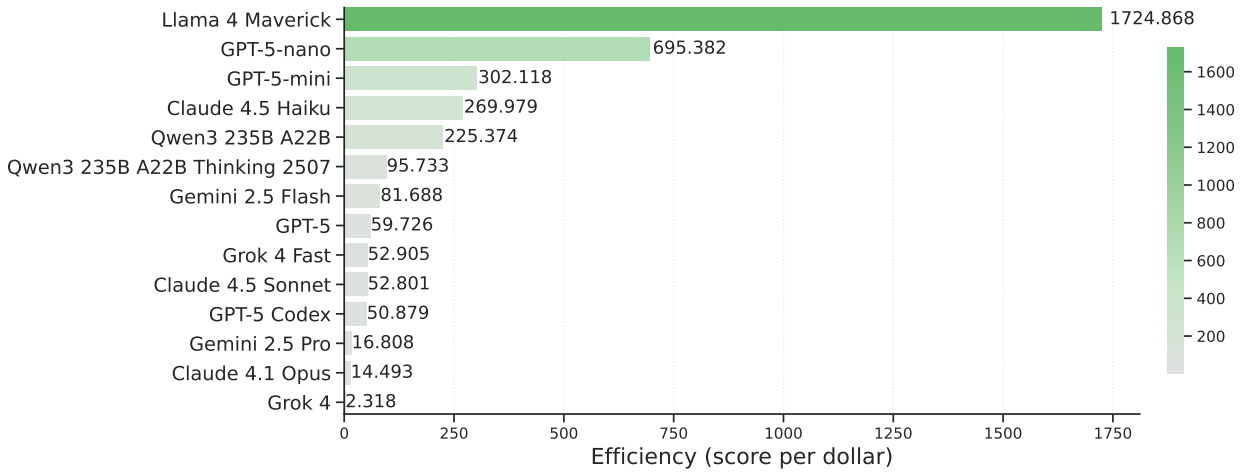


(c) Comparison of performance–time–cost Efficiency Scores **(input modality: Text+Figure, prompt language: English)**

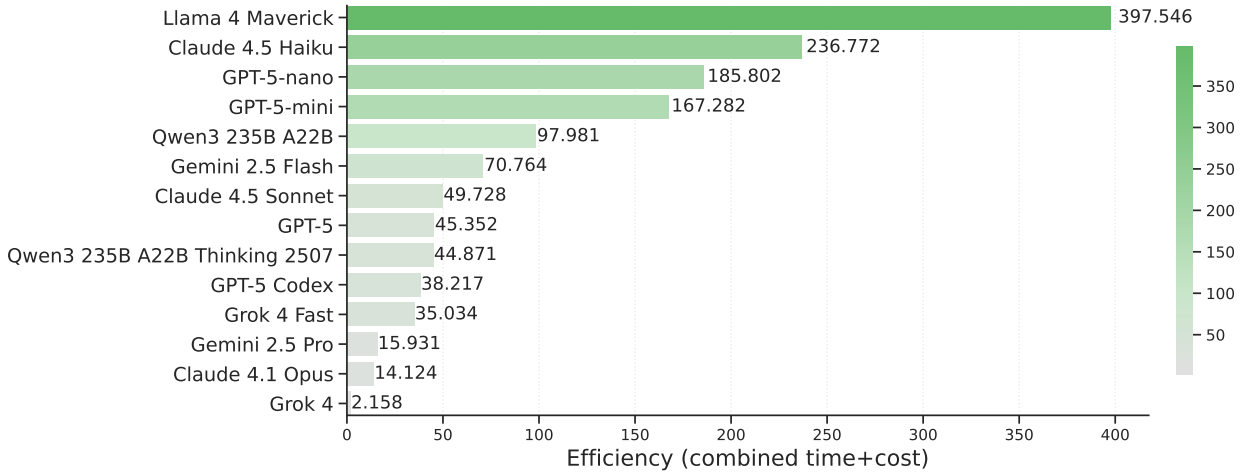
Figure 22: Comparison of performance–time–cost Efficiency Scores across models **(input modality: Text+Figure, prompt language: English)**



(a) Comparison of performance–time Efficiency Scores **(input modality: Image-only, prompt language: English)**



(b) Comparison of performance–cost Efficiency Scores **(input modality: Image-only, prompt language: English)**



(c) Comparison of performance–time–cost Efficiency Scores **(input modality: Image-only, prompt language: English)**

Figure 23: Comparison of performance–time–cost Efficiency Scores across models **(input modality: Image-only, prompt language: English)**

C Full Results by Model Size: Performance and Latency

This appendix reports the full results on model size, extending the analysis in Section 5.2.1 to all input–language combinations (Text / Image / Text+Figure $\times$ Korean / English). Figure 24 compares performance (Normalized Score) and average time per problem for the (Text+Figure, Korean) condition; Figure 25 for the (Image-only, Korean) condition; Figure 26 for the (Text-only, English) condition; Figure 27 for the (Text+Figure, English) condition; and Figure 28 for the (Image-only, English) condition. Models that do not support image input are excluded from the Image-only and Text+Figure conditions. The corresponding plots for the (Text-only, Korean) condition are provided in Figure 12 in the main text.

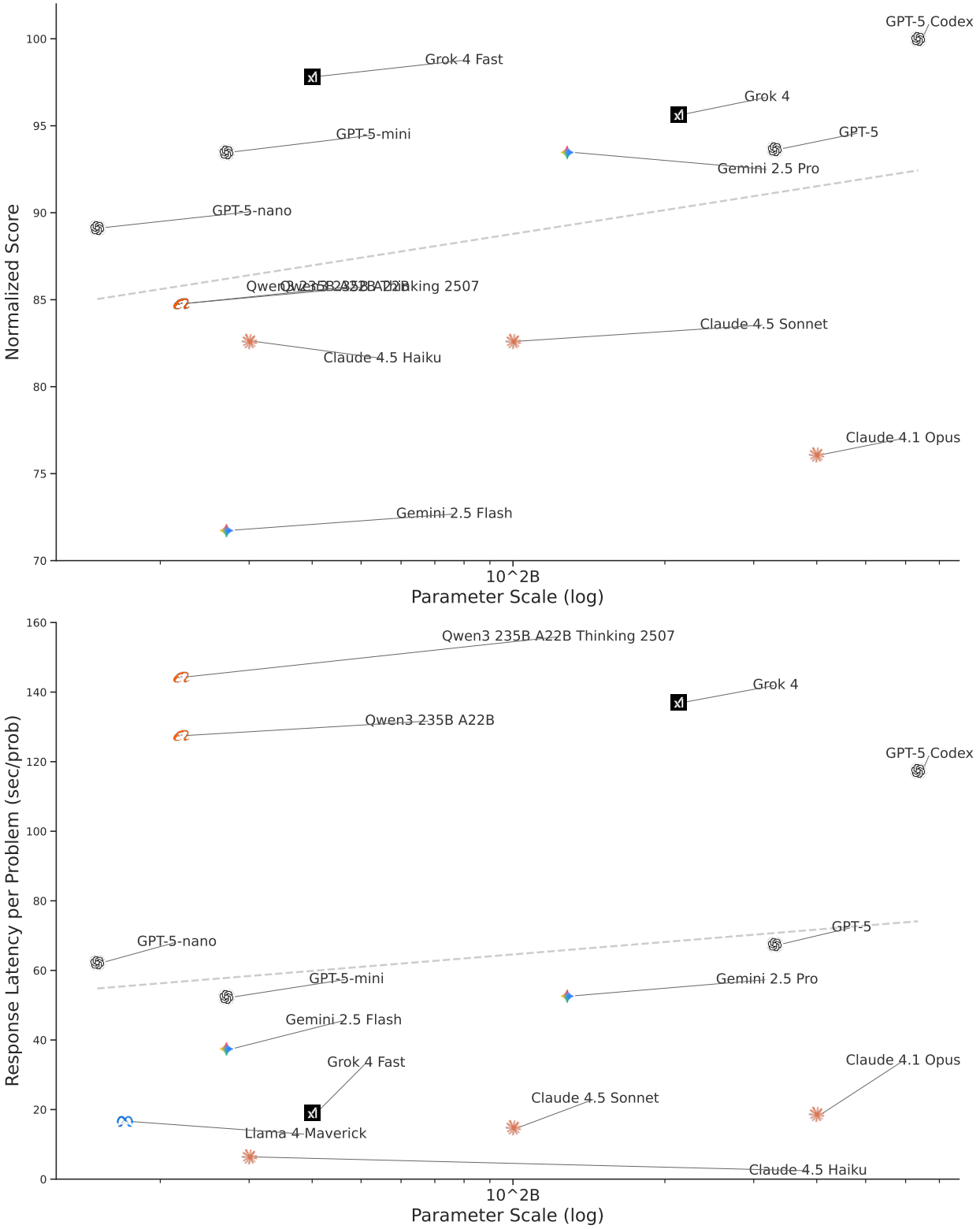


Figure 24: Performance (Normalized Score) and average time per problem by model parameter size **(input modality: Text+Figure, prompt language: Korean)**. The Llama 4 Maverick model is omitted because its Normalized Score (28.3) is substantially lower than other models.

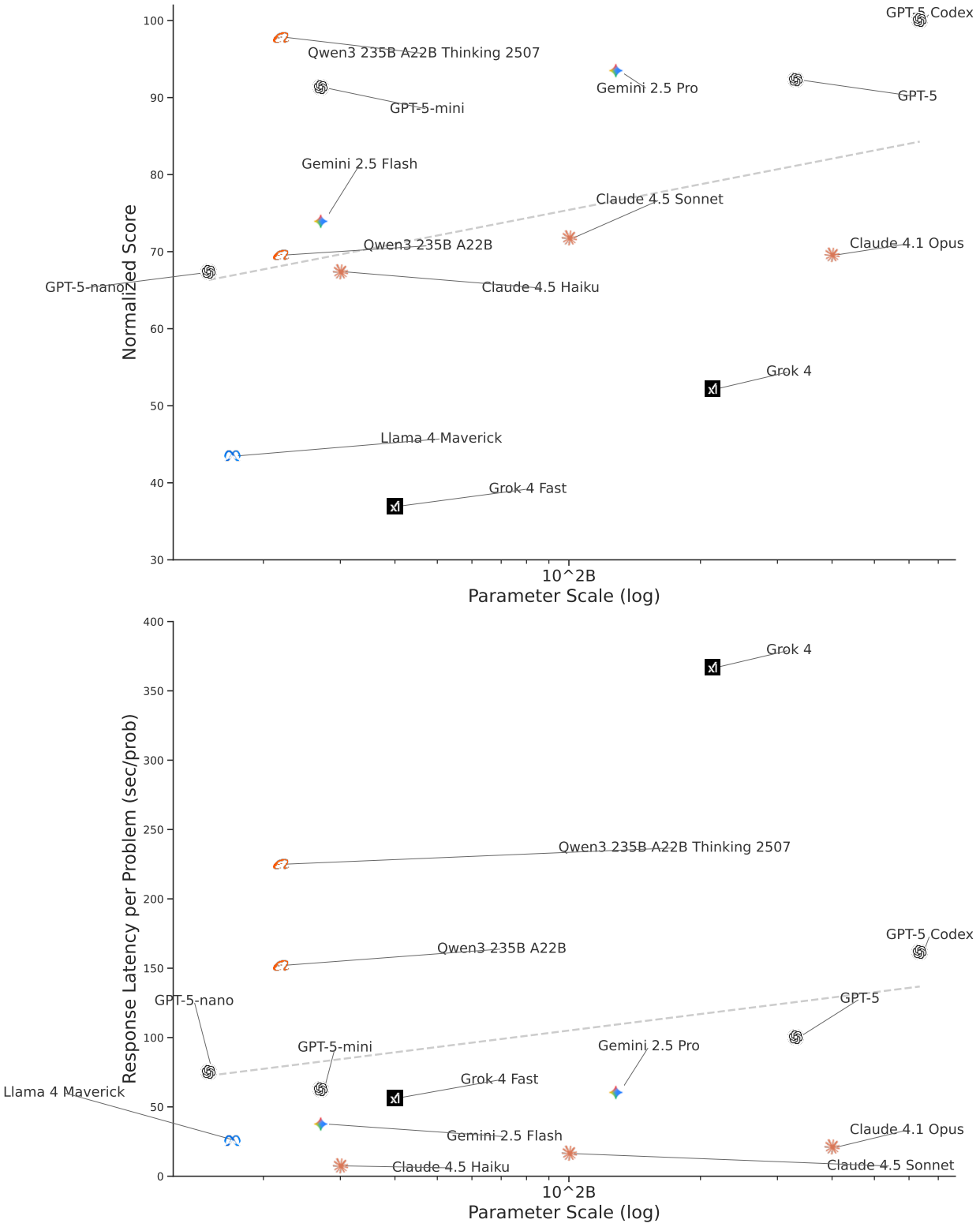


Figure 25: Performance (Normalized Score) and average time per problem by model parameter size **(input modality: Image-only, prompt language: Korean)**

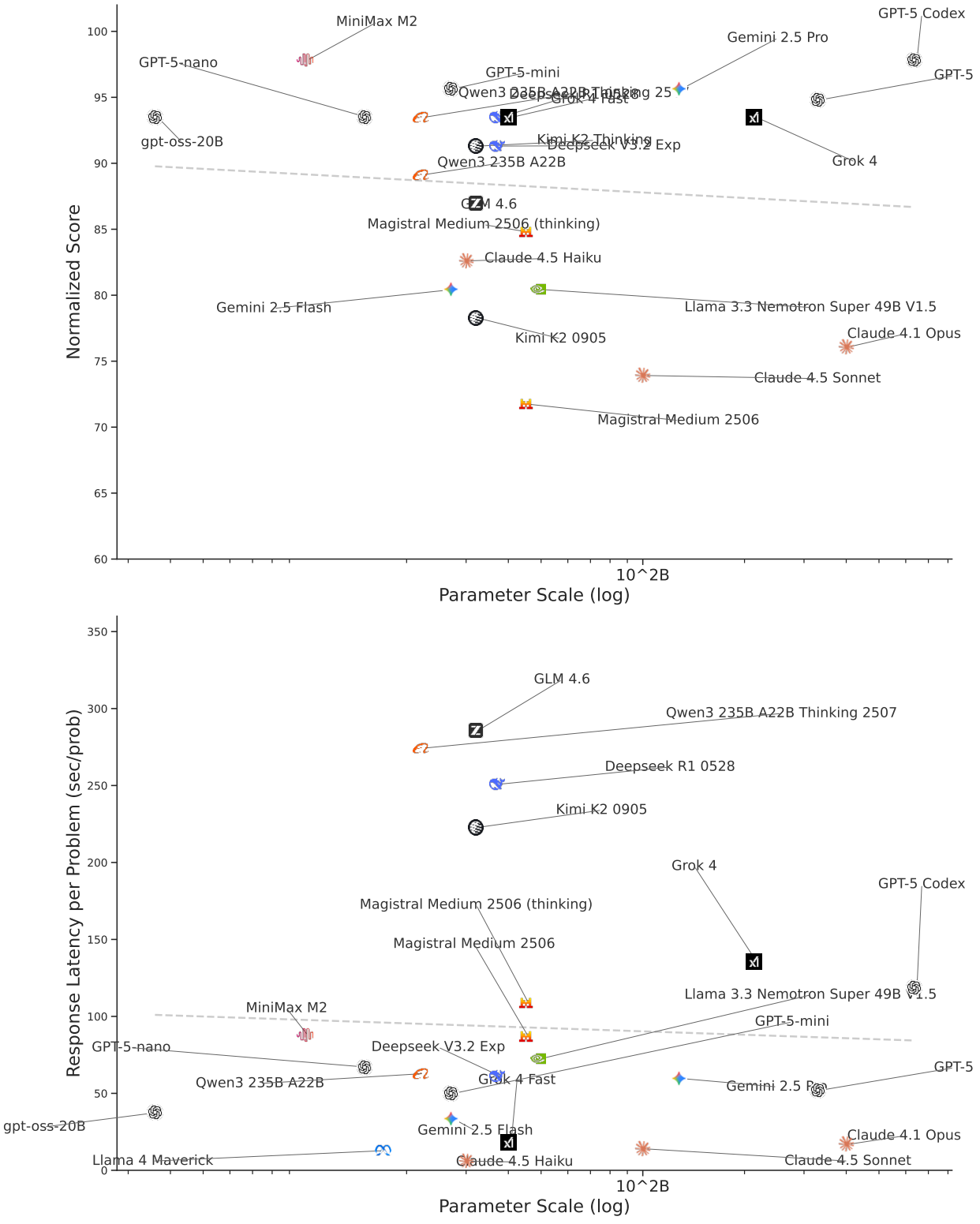


Figure 26: Performance (Normalized Score) and average time per problem by model parameter size **(input modality: Text-only, prompt language: English)**. The Llama 4 Maverick model is omitted because its Normalized Score (23.9) is substantially lower than other models, and the Kimi K2 Thinking model is omitted because its average time per problem (570 sec/prob) is much higher than the others.

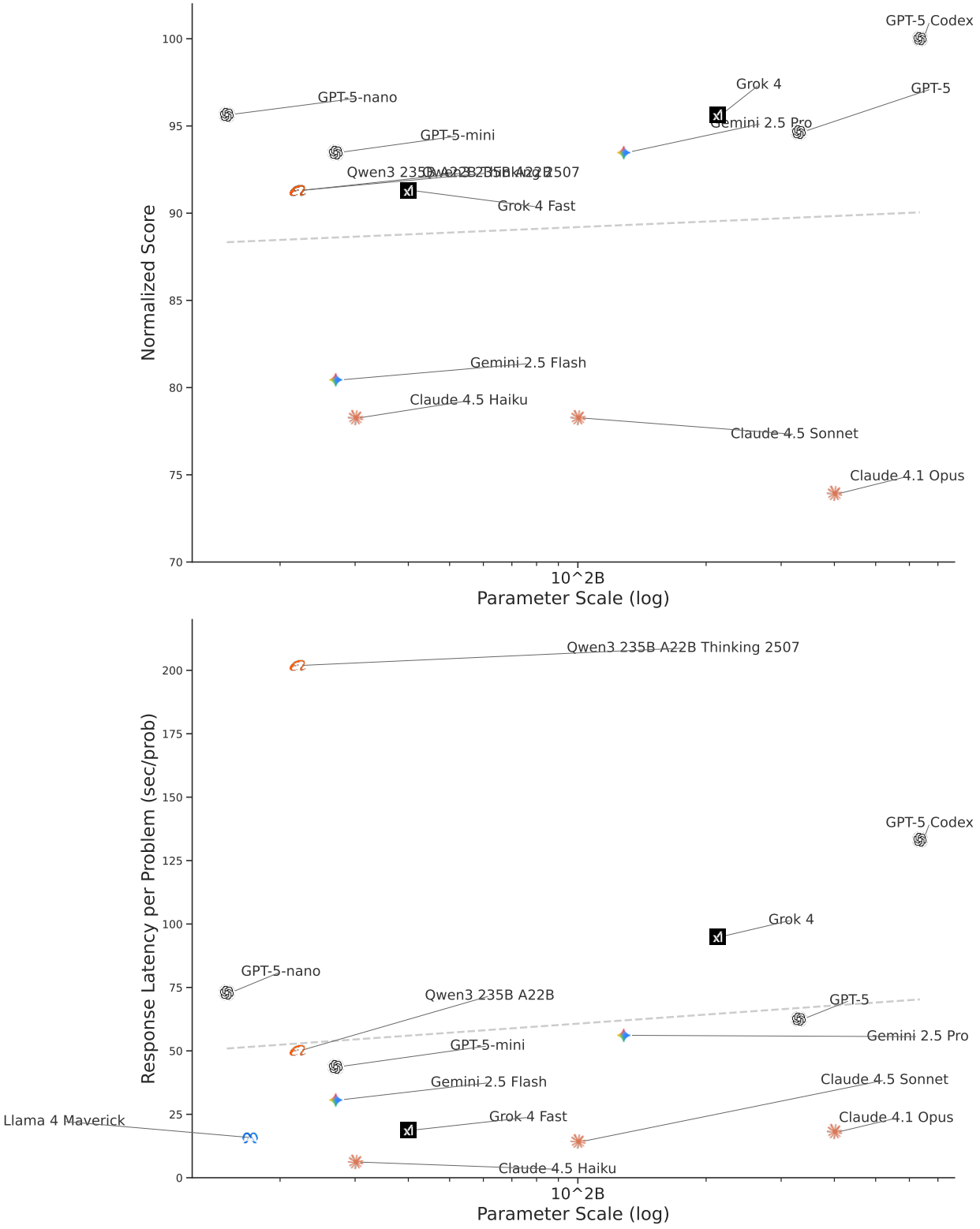


Figure 27: Performance (Normalized Score) and average time per problem by model parameter size **(input modality: Text+Figure, prompt language: English)**. The Llama 4 Maverick model is omitted because its Normalized Score (32.6) is substantially lower than other models.

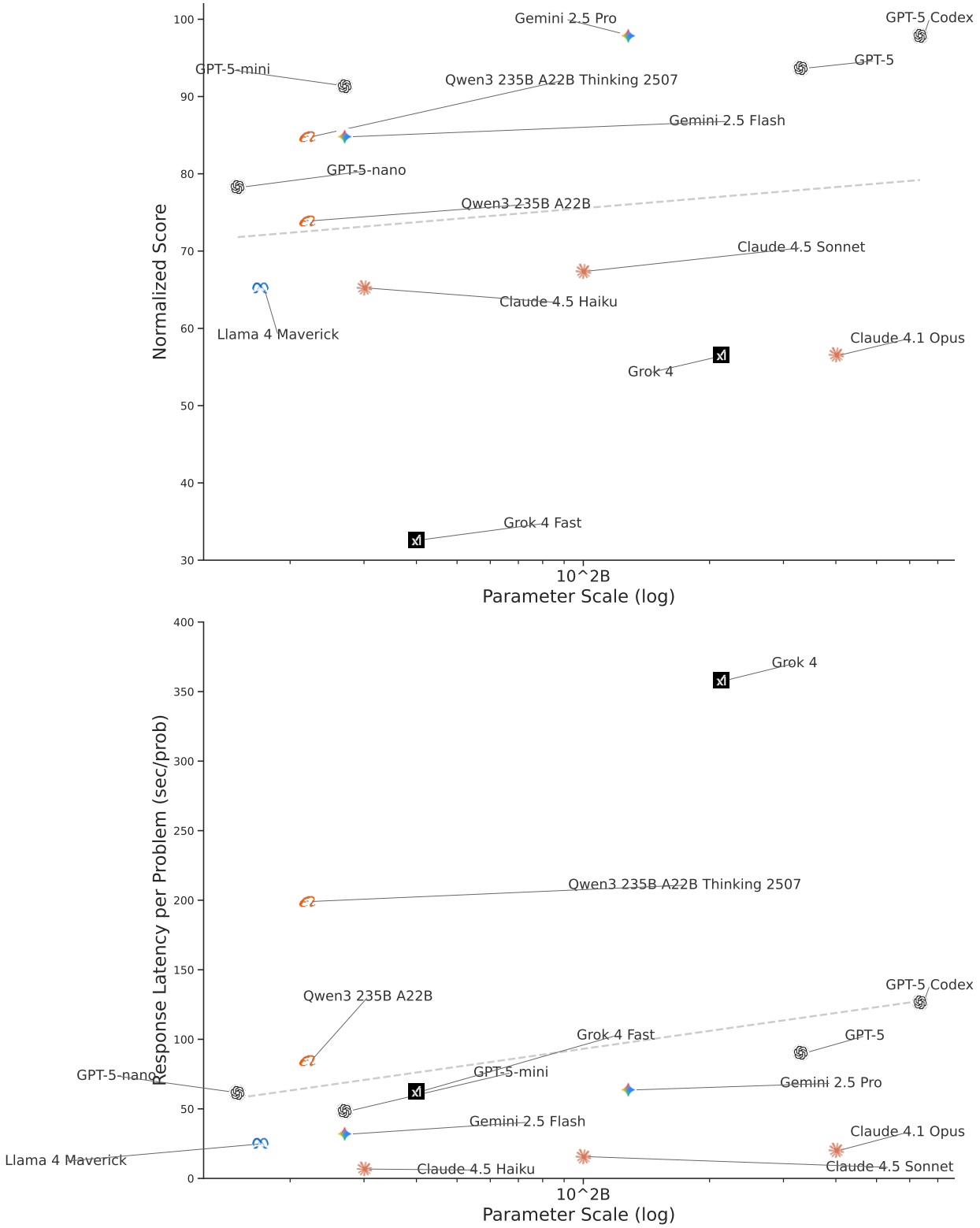


Figure 28: Performance (Normalized Score) and average time per problem by model parameter size **(input modality: Image-only, prompt language: English)**

D Full Results by Problem Area

This appendix provides the complete results by problem area (Common Mathematics, Probability and Statistics, Calculus, Geometry), extending the analysis in Section 5.3.1. For every input–language combination (Text / Image / Text+Figure × Korean / English), we compute the normalized score for each problem area. These results allow a finer examination of whether a model's strengths and weaknesses appear consistently across modalities, or whether the patterns shift depending on input format.

Figure 29 presents area-wise results for the (Text+Figure, Korean) condition; Figure 30 for the (Image-only, Korean) condition; Figure 31 for the (Text-only, English) condition; Figure 32 for the (Text+Figure, English) condition; and Figure 33 for the (Image-only, English) condition. Models without image-input capability are excluded from Image-only and Text+Figure evaluations. Results for the (Text-only, Korean) condition appear in Figure 15 in the main text.

Figure 29: Comparison of LLM performance by problem area **(input modality: Text+Figure, prompt language: Korean)**. Normalized scores represent the ratio of points earned to the maximum points for each area. Both the normalized score and raw points are shown. Models are ordered by overall Normalized Score in descending order.

Figure 30: Comparison of LLM performance by problem area **(input modality: Image-only, prompt language: Korean)**. Normalized scores represent the ratio of points earned to the maximum points for each area, with raw points displayed. Models appear in descending order of overall Normalized Score.

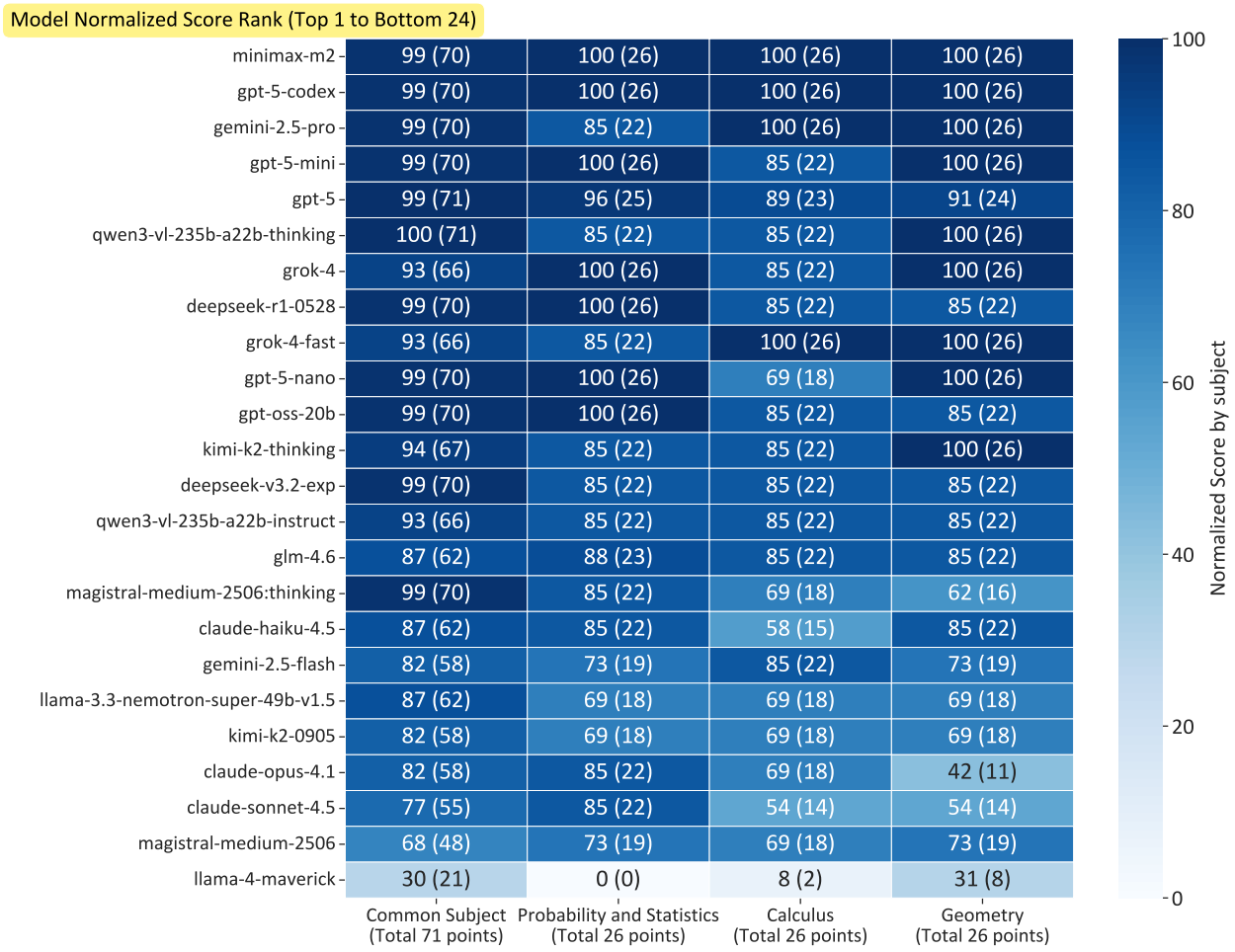


Figure 31: Comparison of LLM performance by problem area **(input modality: Text-only, prompt language: English)**. Normalized scores are computed using area-specific maximum scores; raw points are shown together. Models are sorted by overall Normalized Score.

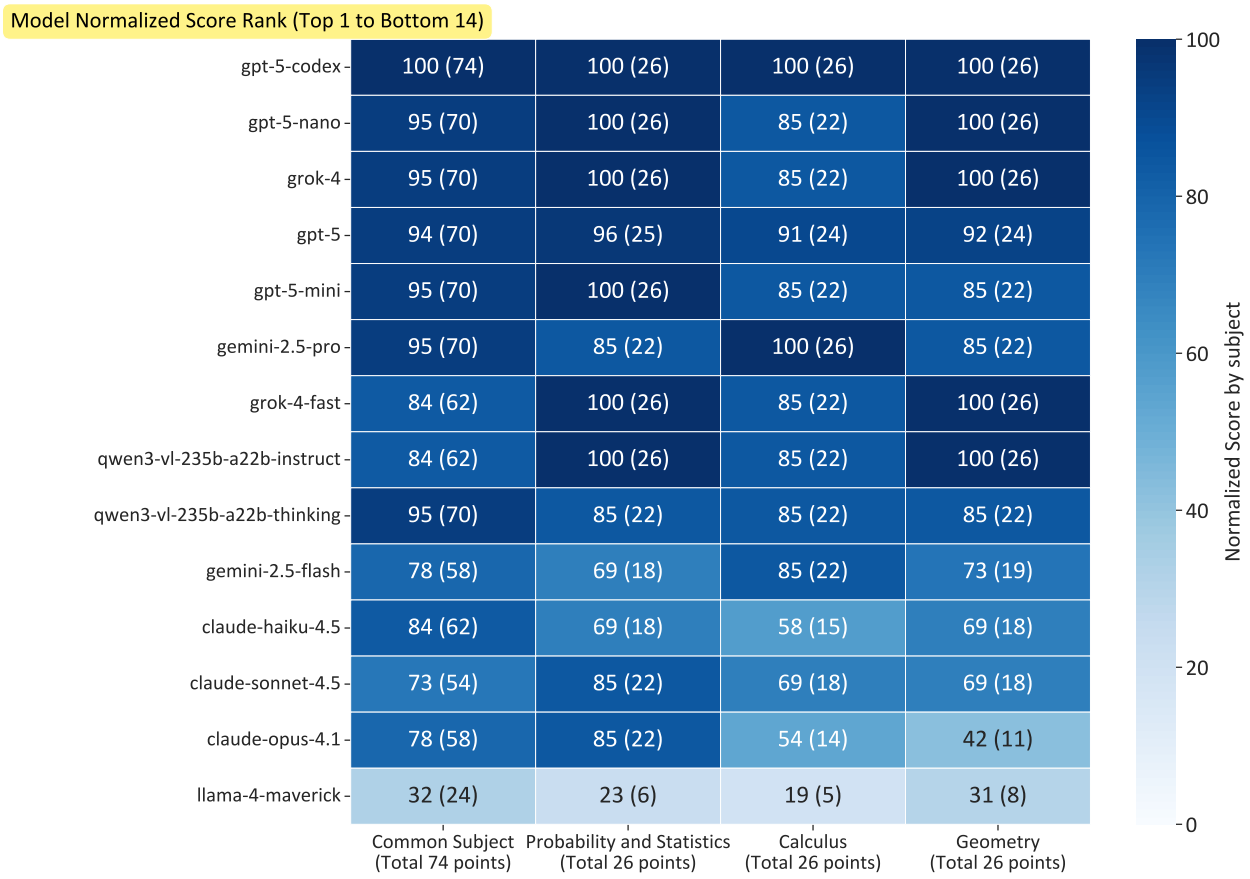


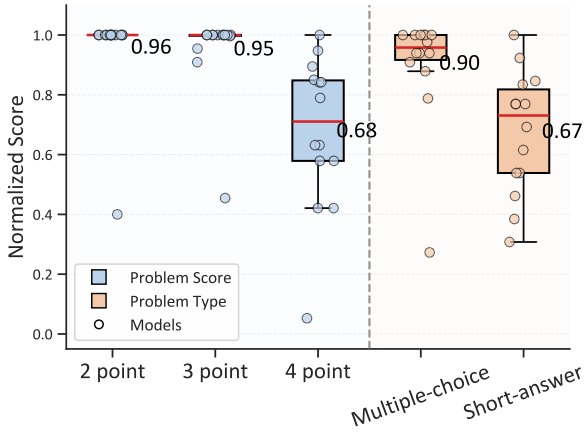
Figure 32: Comparison of LLM performance by problem area **(input modality: Text+Figure, prompt language: English)**. Normalized scores represent the ratio of earned to maximum points, with raw points included. Models are ordered by overall Normalized Score.

Figure 33: Comparison of LLM performance by problem area **(input modality: Image-only, prompt language: English)**. Normalized scores and raw points are shown. Models appear in descending order of overall Normalized Score.

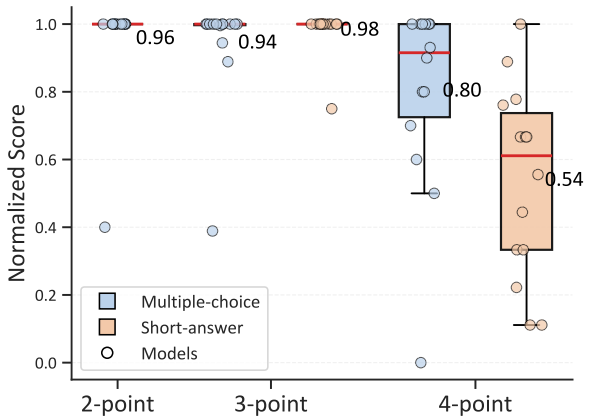
E Full Results by Problem Attributes

This appendix extends the analysis in Section 5.3.2 by reporting full results across problem attributes (problem format and point value) and all input–language combinations (Text / Image / Text+Figure × Korean / English).

Figure 34 presents the results for (Text+Figure, Korean); Figure 35 for (Image-only, Korean); Figure 36 for (Text-only, English); Figure 37 for (Text+Figure, English); and Figure 38 for (Image-only, English). The corresponding (Text-only, Korean) figure appears in Figure 16 in the main text.

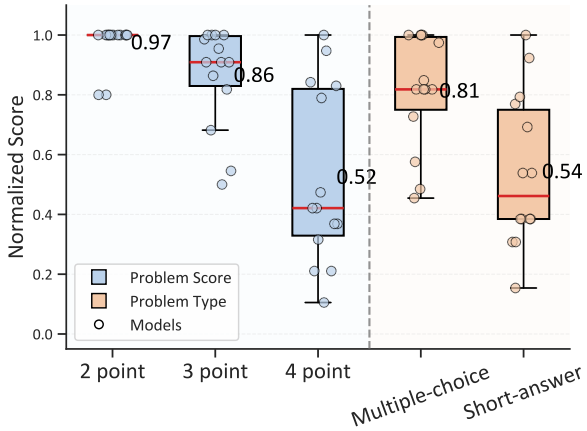


(a) Distribution of LLM performance by point value (2/3/4 points) and problem format (multiple-choice / short-answer).

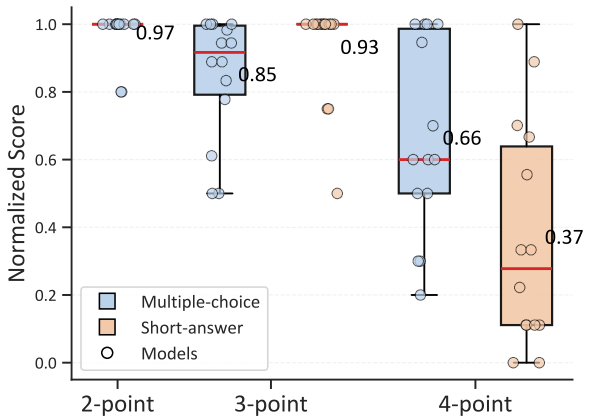


(b) LLM performance by combined problem attributes (point value × problem format). (The 2026 CSAT contains no 2-point multiple-choice problems.)

Figure 34: Comparison of LLM performance by problem attributes **(input modality: Text+Figure, prompt language: Korean)**. Panel (a) examines point value and problem format independently, while panel (b) analyzes performance by combined problem categories (point value × format).

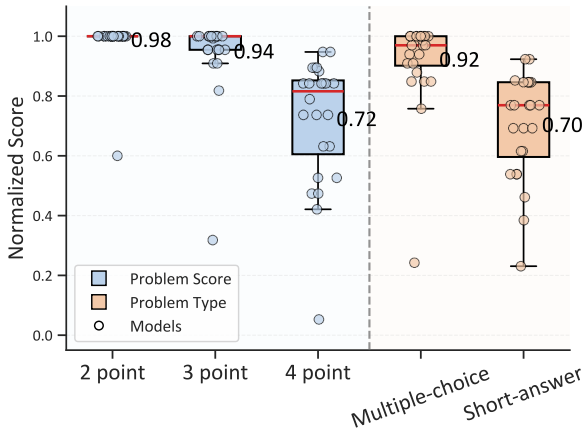


(a) Distribution of LLM performance by point value and problem format.

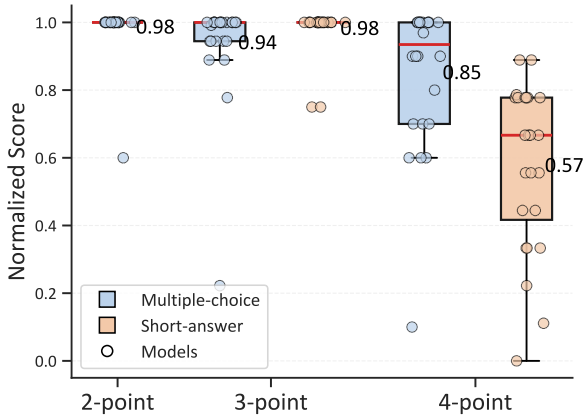


(b) LLM performance by combined problem attributes (point value × format). (No 2-point multiple-choice problems in the 2026 CSAT.)

Figure 35: Comparison of LLM performance by problem attributes **(input modality: Image-only, prompt language: Korean)**. Panel (a) isolates the two attributes, while panel (b) evaluates combined problem categories.

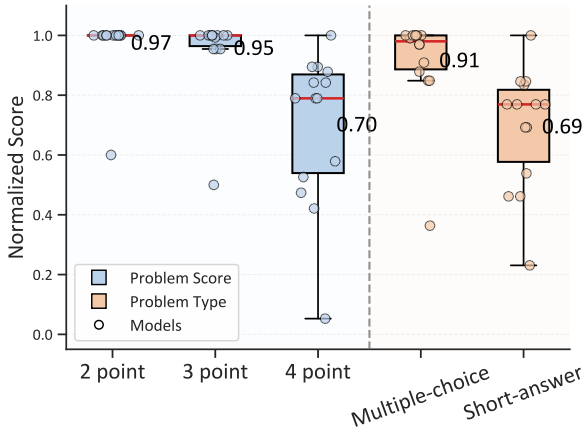


(a) Distribution of LLM performance by point value and problem format.

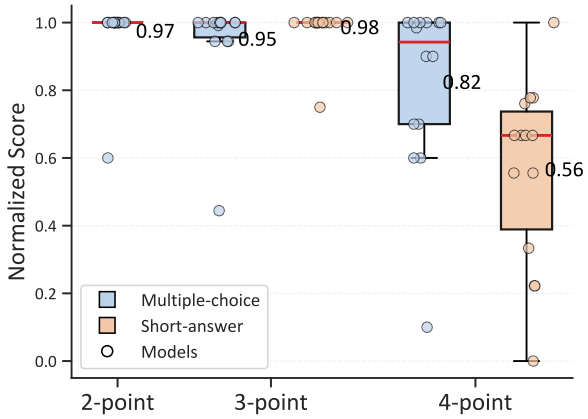


(b) LLM performance by combined problem attributes (point value × format). (No 2-point multiple-choice problems in the 2026 CSAT.)

Figure 36: Comparison of LLM performance by problem attributes **(input modality: Text-only, prompt language: English)**. Panel (a) examines problem format and point value separately, whereas panel (b) analyzes their combined categories.

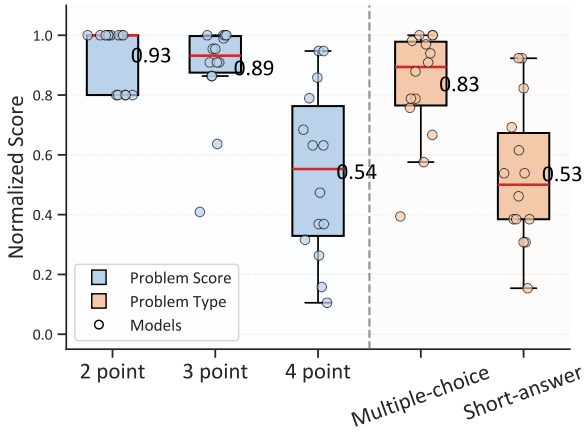


(a) Distribution of LLM performance by point value and problem format.

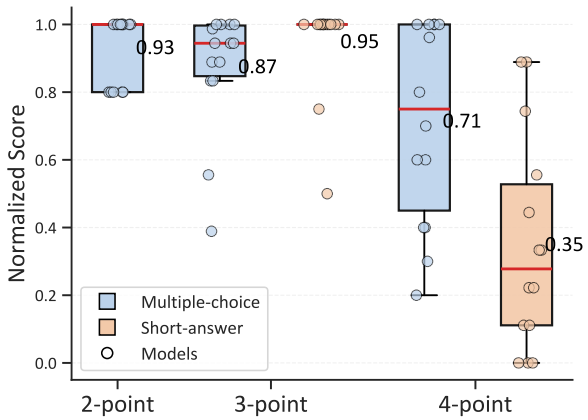


(b) LLM performance by combined problem attributes (point value × format). (No 2-point multiple-choice problems in the 2026 CSAT.)

Figure 37: Comparison of LLM performance by problem attributes **(input modality: Text+Figure, prompt language: English)**. Panel (a) analyzes attributes independently, while panel (b) examines combined problem categories.



(a) Distribution of LLM performance by point value and problem format.



(b) LLM performance by combined problem attributes (point value × format). (No 2-point multiple-choice problems in the 2026 CSAT.)

Figure 38: Comparison of LLM performance by problem attributes **(input modality: Image-only, prompt language: English)**. Panel (a) shows separate attribute-wise performance, while panel (b) presents performance for combined problem categories.