SCI-ρ: A MULTILINGUAL VISUALLY-GROUNDED SYMBOLIC BENCHMARK FOR STEM PROBLEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Symbolic benchmarks have emerged as a key approach to assess model robustness under minor modifications to STEM-related questions. However, existing symbolic benchmarks mostly remain limited to mathematical reasoning, lack visual grounding, and are predominantly in English. In this work, we introduce Sci-ρ (**Sci**ence **Rho**bustness), a dynamic benchmark for visually-grounded STEM problems spanning five subjects and seven languages, comprising 4,242 problem templates (606 per language) crafted by domain experts, including Olympiad medalists. Each template is implemented as *executable Python code* that generates diverse but equivalent problem instances by varying numerical values, visual patterns, geometric shapes, color schemes, and function types, resulting in 42,420 instances in total, each paired with reasoning steps and ground-truth solutions. We evaluated 17 state-of-the-art VLMs and discovered a noticeable gap between worst-case accuracy (defined as the proportion of problem templates that a model answers correctly across every generated variation) and average accuracy. We also discovered that smaller models show noticeable performance degradation across languages, whereas proprietary and larger models remain robust. Step-level evaluation reflects this same trend, revealing a significant gap between average F1 and worst-case F1 scores. Finally, our inspection of attention heads of a VLM reveals substantial cross-lingual variation in the relative attention allocated to image tokens compared to text tokens. Our work highlights the importance of evaluation beyond static benchmarks as a metric to measure the quality of VLMs.

1 INTRODUCTION

Large Vision-Language Models (VLMs) have demonstrated notable capabilities in solving visually-grounded tasks across various domains (Ren et al., 2025; Shinoda et al., 2025; Nayak et al., 2024; Nyandwi et al., 2025). To measure progress on reasoning that requires visual understanding, the community has introduced a wide range of benchmarks: `MATH-V` (Wang et al., 2024), `MathVista` (Lu et al., 2024), and `MM-Math` (Sun et al., 2024) for Mathematics; `SeePhys` (Xiang et al., 2025), `PHYBench` (Qiu et al., 2025), and `PhysReason` (Zhang et al., 2025) for Physics; and `SCIENCEQA` (Lu et al., 2022) and `SciVerse` (Guo et al., 2025) for more general science questions. Across these benchmarks, VLMs consistently fall short of human performance.

Although these benchmarks represent an important early effort, the robustness of VLM problem-solving remains largely unexplored. Existing evaluations mostly rely on static, one-off images: each problem is rendered exactly once, and the model is scored on that single instance. Yet a problem's visual presentation is incidental to its underlying logic–the same geometric relationship, for instance, can be drawn with different side lengths without altering the reasoning required to solve it. Humans solve such variants without difficulty; whether VLMs do is an open question. Without controlled perturbations of the image itself, it is impossible to distinguish models that genuinely reason about visual content from those that succeed by matching surface features of problems seen during training.

Figure 1 shows an example where GPT-5.4—a frontier model—successfully answers a problem in one scenario but fails in another seemingly similar question where the graph and point coordinates are varied. This inconsistency suggests a lack of robust reasoning capability and highlights the importance of evaluation beyond static benchmarks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

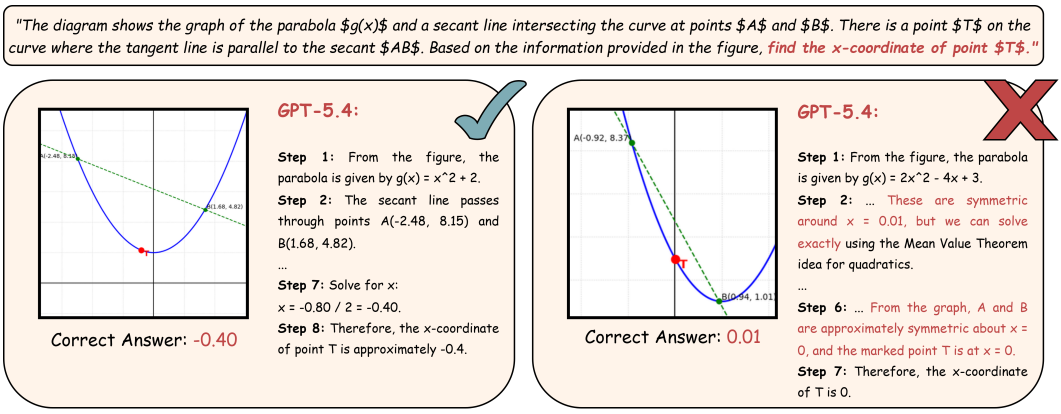


Figure 1: An illustrative failure case of GPT-5.4 when evaluated across problem variants in our dataset, highlighting vulnerabilities in VLM robustness.

A further limitation of prior work is its narrow scope. Coverage is concentrated on Mathematics and Physics, leaving the rest of STEM–Biology, Chemistry, and Computer Science–largely outside the scope of visually-grounded evaluation, despite each presenting distinct visual reasoning demands such as molecular structures, biological diagrams, and algorithmic traces. Evaluation is also conducted almost entirely in English, leaving open whether observed capabilities transfer to the multilingual settings in which most learners and practitioners actually encounter STEM content.

We address these gaps by introducing the first symbolic, multilingual robustness benchmark for visually-grounded STEM reasoning. Our benchmark spans Mathematics, Physics, Biology, Chemistry, and Computer Science at the high school and pre-university level, with each problem implemented as a Python template that dynamically generates question variants—along with their reasoning steps and solutions—across seven languages. These templates were carefully constructed and reviewed by domain experts, including science olympiad medalists, degree holders, and final-year undergraduates with demonstrated expertise in their respective fields. Our contributions are as follows:

- We developed Sci-ρ (**Sci**ence **Rho**bustness) as a multilingual multimodal symbolic benchmark to assess VLMs capabilities in solving STEM-Problems. The dataset consists of $42,420$ problem instances generated from 606 templates, 7 languages, and 10 variants in five core STEM subjects: Mathematics, Physics, Chemistry, Biology, and Computer Science.
- We benchmarked 17 VLMs using Sci-ρ . Our evaluation provides an in-depth analysis of the performance gap between worst-case and average accuracy, the cross-lingual performance degradation relative to English, and the step-level validity of the models' reasoning compared to ground-truth solutions.
- We perform a mechanistic analysis of how VLMs allocate attention between visual and textual content during STEM reasoning. We define the Image Attention Ratio (IAR) to quantify the magnitude of attention paid to image tokens relative to textual tokens, and compare IAR across languages and subjects on a representative open-weight VLM.

2 Related Works

Visual Language Models Vision-Language Models (VLMs) have advanced at a remarkable pace, with proprietary models like GPT-5 (Singh et al., 2025) and Gemini 2.5 (Google DeepMind, 2025b) setting the state of the art. The open-source community has kept pace through diverse architectural innovations: Gemma 3's optimized local-to-global attention ratios (Team et al., 2025), Qwen3's multi-level ViT feature integration, and the native multimodal capabilities of Qwen 3.5. Together with models such as Llama 4 Scout (Meta AI, 2025), InternVL 3.5 (Wang et al., 2025), SEA-LION VL (AI Singapore, 2026), and Molmo 2 (Clark et al., 2026), these systems continue to redefine performance on visual benchmarks. Yet robustness remains a critical concern (Vo et al., 2026; Irawan

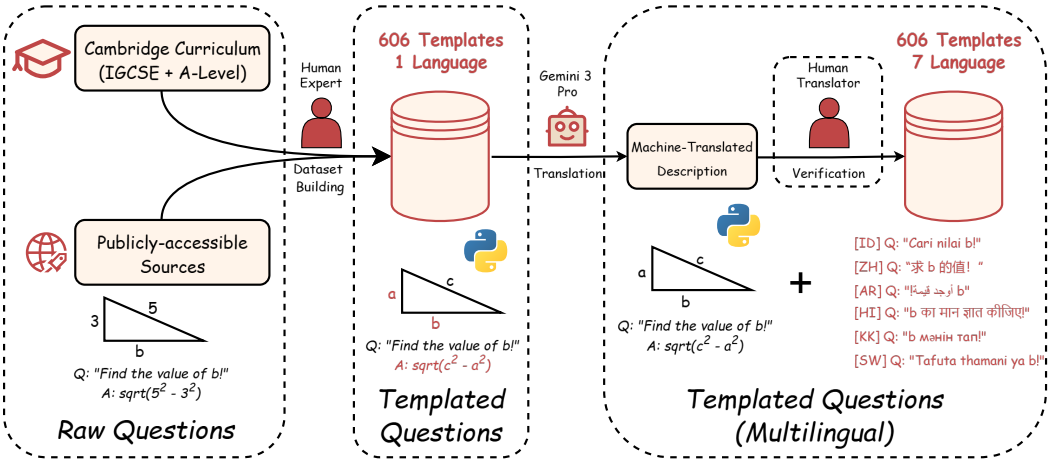


Figure 2: Dataset construction pipeline.

et al., 2026): persistent failures on simple tasks such as object counting raise questions about the sufficiency of static benchmarks, particularly for rigorous STEM applications.

Robustness & Symbolic Benchmark Considerable research effort has been directed toward evaluating the robustness of LLMs and VLMs. Zheng et al. (2024) found that LLMs are vulnerable to changes in option position when solving Multiple Choice Questions due to their inherent selection bias. Another work by Zong et al. (2024) discovered a similar option-shuffling vulnerability in VLMs. Recently, Mirzadeh et al. (2025) discovered that LLMs show a non-negligible variance when solving the same textual mathematical question with different numerical values, indicating that current LLMs are not capable of genuine logical reasoning. Extensive study on VLMs by Zou et al. (2025) concluded that VLMs are sensitive to changes in input variations when solving math problems, despite the knowledge and steps required to solve the problem remaining the same.

Yet across this body of work, robustness evaluation through symbolic, programmatically generated variants for visually-grounded reasoning has been confined almost exclusively to mathematics and to English. To our knowledge, no prior benchmark provides symbolic, image-level perturbations for visually-grounded questions in Physics, Chemistry, Biology, or Computer Science, nor in any non-English language. Our work is the first to address both gaps in a single benchmark.

3 SCI-ρ DATASET

3.1 DATASET CONSTRUCTION

Figure 2 illustrates the dataset construction pipeline, which was conducted in two stages: template creation and translation. In the first stage, we collected vision-dependent STEM problems from open sources and transformed them into dynamic Python classes, guided by Cambridge curriculum. In the second stage, we translated the problems into six additional languages with verification by native speakers.

Template Creation We curated a list of vision-dependent materials based on the Cambridge curriculum as our ground reference[1]. This includes several topics from the Cambridge IGCSE and A-level curricula, spanning five subjects: Mathematics, Physics, Chemistry, Biology, and Computer Science. After identifying the relevant topics and subtopics to be included in our dataset, we collected problems from various publicly available sources, mostly from Cambridge past paper exams (IGCSE & A-Level), previous benchmark, and originally-constructed questions, and subsequently assigned each problem to the appropriate topic and subtopic defined earlier. We only included problems that depends on image to solve, as not all problems in STEM require visual component.

[1]We use the Cambridge curriculum as a high-level guideline to structure topics and subtopics. However, we do not strictly follow this curriculum, and the dataset includes additional topics and problem types beyond it.

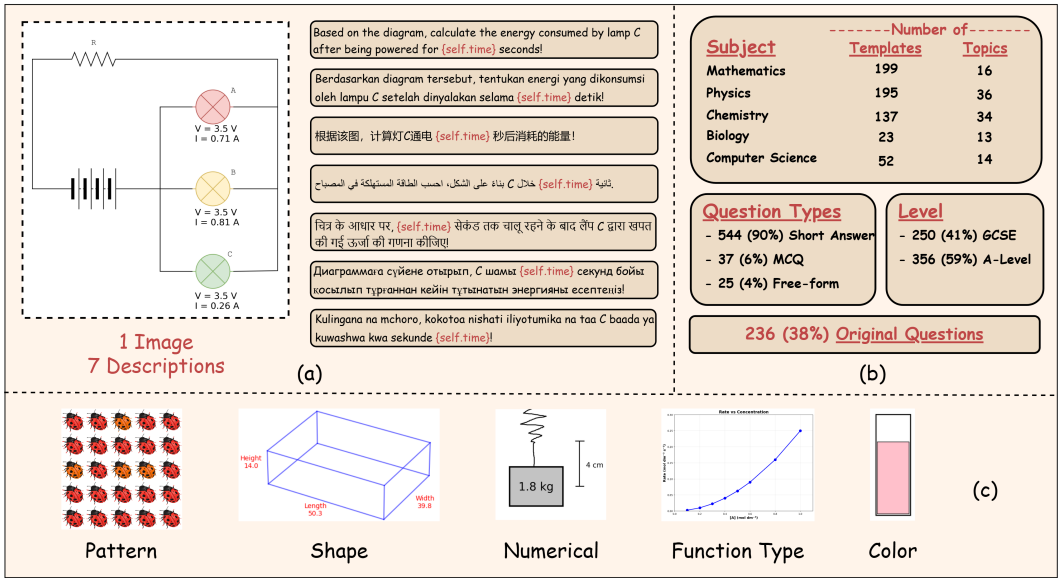


Figure 3: A glimpse of our dataset. (a) An example from our dataset where each template consists of one image and description in seven languages; (b) Statistics of our dataset; (c) Example of variants included in our dataset.

We hired STEM graduates, advanced undergraduates, and science olympiad medalists as expert workers, and tasked them with manually rewriting each question into a Python class that serves as a dynamic template. For each template, we included several attributes critical to the problem (such as description, image, final answer) along with common metadata (topic, subtopic, source, curriculum, and question type). All calculations leading to the final answer were also computed and stored in each Python class. To provide a clear trace of the reasoning process, our dataset includes dynamically generated step-by-step solutions written in LaTeX. While there is no theoretical upper limit to the number of variants a template can generate, each is guaranteed to yield at least ten distinct problem instances.

Finally, we applied strict quality control. Problem setters first cross-checked each other's work within the same subject, after which a dedicated team of quality control workers manually verified the outputs generated from each Python template, examining both the intermediate reasoning steps and the final answer for correctness. Further details are provided in Appendix A.2. This stage yielded 606 high-quality multimodal problem templates in English.

Translation In this step, the dataset was expanded into six additional languages: Indonesian, Chinese, Arabic, Hindi, Kazakh, and Swahili. We first translated the descriptions for each template using Gemini 3 Pro while keeping the images in English. These machine-translated descriptions were then verified by native speakers of each corresponding language, involving both the authors and hired contributors. We also employed quality control by intentionally introducing errors into 40 of the 606 descriptions (approximately 7%) at random. Translators who failed to identify and correct these errors were required to revise their work (more details are provided in Appendix A.3). Finally, we integrated these translations into our main dataset. The result is 606 high-quality problem templates in seven languages with localized descriptions and a unified English visual component.

3.2 Dataset Overview

Our dataset consists of 606 high-quality templates written in Python, encompassing five STEM subjects (Mathematics, Physics, Chemistry, Biology, and Computer Science) across seven language variants (English, Indonesian, Chinese, Arabic, Hindi, Kazakh, and Swahili). Each template is categorized into exactly one topic and one subtopic, taking the form of a (numerical) Short Answer, MCQ, or Free-form problem, and incorporates several dimensions of variability.

Figure 3 (part a) shows a sample taken from the Physics subset of our dataset, which consists of descriptions in seven languages and one unified image. Specifically for this template, both the information shown in the image, such as voltage (V) and current (I), and the variable *self.time* in the description are dynamic. This setup allows us to evaluate performance variations when a VLM is prompted with diverse numerical information across different languages.

The statistics of our dataset are presented in Figure 3 (part b). In general, our dataset is predominantly composed of Mathematics, Physics, and Chemistry problems, comprising 199, 195, and 137 templates respectively, followed by a smaller number of Computer Science (52) and Biology (23) templates. The number of topics for each subject is given in the figure. The majority of the dataset consists of Short Answer problems (accounting for 90% of the total), followed by a smaller proportion of MCQs (6%) and Free-form questions (4%). Our dataset spans two difficulty levels, with more than half (59%) of the problems being at the A-Level and 41% at the GCSE level. About 38% of the problems are original (designed from scratch), while the remainder are adapted from various open sources.

We identify five dimensions of variability in our dataset, as illustrated in Figure 3 (part c). The *pattern* and *color* variants test the model capabilities to interpret positional variants (such as the arrangement of ladybugs) and diverse color schemes (such as the color of a chemical solution). To assess spatial and structural comprehension, the *shape* variant introduces different geometric forms, such as varying cuboid dimensions. Furthermore, to probe quantitative and analytical capabilities, the *numerical* variant challenges models by varying values embedded within either the text or the accompanying image, while the *function type* variant measures proficiency in distinguishing mathematical functions (e.g., linear vs quadratic). More examples are provided in Appendix B.

4 EXPERIMENT

We evaluated Vision Language Models using six metrics: Average-case Accuracy ($\mathcal{A}_{avg}$), Worst-case Accuracy ($\mathcal{A}_{wst}$), Step Recall ($\mathcal{R}$), Step Precision ($\mathcal{P}$), Step F1 ($\mathcal{F}$), and Worst-case Step F1 ($\mathcal{F}_{wst}$), defined in Section 4.1. Additionally, we analyzed the attention heads of *Qwen3-VL-8B-Instruct* to better understand how strongly the model attends to image tokens compared to text tokens.

4.1 EVALUATION SETUP

Our dataset consists of 606 problem templates. From each template, we generated $M = 10$ distinct problem instances across 7 languages, yielding a total of $10 \times 606 \times 7 = 42,420$ unique problem instances. This comprehensive collection was then utilized to rigorously evaluate VLM robustness.

For our experiments, we benchmarked leading closed-weight models, including Gemini 2.5 Pro (Google DeepMind, 2025b), Gemini 2.5 Flash (Google DeepMind, 2025a), and GPT-5.4 (Singh et al., 2025), alongside open-weight models such as Llama 4 Scout (Meta AI, 2025), Qwen VL(3 & 3.5) (Bai et al., 2025), Gemma 3 (Team et al., 2025), InternVL 3.5 (Wang et al., 2025), SEA-LION VL (AI Singapore, 2026), and Molmo 2 (Clark et al., 2026). We employed prompt engineering to instruct the model to generate responses in JSON format, which consists of step-by-step solution and a concise final answer. We subsequently parsed the JSON outputs to extract both the reasoning trace and the final answer. In cases when a model failed to generate the requested format, we implemented regex-based pattern matching as a fallback parsing mechanism. To account for minor deviations in numerical short answers, we accept values within a 1% tolerance of the ground truth. More details regarding the experiment setup are available in Appendix C.

To evaluate model performance, we adopt metrics introduced by Zou et al. (2025): **average-case accuracy** ($\mathcal{A}_{avg}$) and **worst-case accuracy** ($\mathcal{A}_{wst}$), which defined as follows:

$$\mathcal{A}_{avg} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}\mathbb{I}[\text{Ans}(i,j) = \text{GT}(i,j)],\ \mathcal{A}_{wst} = \frac{1}{N}\sum_{i=1}^{N}\min_{j\in[1,M]}\mathbb{I}[\text{Ans}(i,j) = \text{GT}(i,j)], \tag{1}$$

where $\text{Ans}(i,j)$ and $\text{GT}(i,j)$ represent the generated answer and the ground truth answer, respectively, for variant j of question i.

Table 1: English-only results on Sci-ρ . For each subject, A_{avg} denotes average-case accuracy and A_{wst} denotes worst-case accuracy. Δ denotes the reduction $A_{wst} - A_{avg}$.

Model	Mathematics			Physics			Chemistry			Biology			Compsci		
	A_{avg}	A_{wst}	Δ	A_{avg}	A_{wst}	Δ	A_{avg}	A_{wst}	Δ	A_{avg}	A_{wst}	Δ	A_{avg}	A_{wst}	Δ
Closed-source Models															
GPT-5.4	90.3	72.4	-17.9	94.6	84.6	-10.0	94.4	78.8	-15.6	72.2	47.8	-24.4	95.4	84.6	-10.8
Gemini-2.5-Pro	86.2	66.3	-19.9	90.4	69.2	-21.2	91.7	70.1	-21.6	71.7	43.5	-28.2	91.2	75.0	-16.2
Gemini-2.5-Flash	85.2	61.8	-23.4	87.4	63.1	-24.3	92.2	67.9	-24.3	51.7	30.4	-21.3	86.9	63.5	-23.4
Open-source Models															
Qwen3.5-122B-A10B	82.1	49.2	-32.9	86.7	49.2	-37.5	88.0	43.1	-44.9	67.8	26.1	-41.7	88.5	57.7	-30.8
Qwen3.5-27B	85.1	60.3	-24.8	87.0	56.4	-30.6	90.9	65.7	-25.2	65.7	39.1	-26.6	89.2	63.5	-25.7
Llama-4-Scout	71.4	33.7	-37.7	78.8	43.1	-35.7	84.7	46.7	-38.0	44.3	8.7	-35.6	68.7	25.0	-43.7
InternVL3.5-14B	58.5	21.1	-37.4	69.2	30.3	-38.9	76.8	35.0	-41.8	30.9	4.3	-26.6	64.6	21.2	-43.4
InternVL3.5-8B	64.8	35.2	-29.6	73.8	45.1	-28.7	79.3	40.9	-38.4	36.5	4.3	-32.2	61.0	25.0	-36.0
InternVL3.5-4B	54.0	21.1	-32.9	65.3	26.7	-38.6	72.6	29.9	-42.7	23.9	0.0	-23.9	55.2	17.3	-37.9
Qwen3-VL-8B-Instruct	61.5	25.6	-35.9	70.3	34.4	-35.9	83.6	47.4	-36.2	53.5	8.7	-44.8	67.1	25.0	-42.1
Qwen3-VL-4B-Instruct	31.7	5.0	-26.7	40.4	6.2	-34.2	35.8	3.6	-32.2	17.8	0.0	-17.8	23.8	3.8	-20.0
Molmo2-8B	46.5	14.6	-31.9	61.9	30.3	-31.6	56.6	24.1	-32.5	26.1	8.7	-17.4	40.8	9.6	-31.2
Molmo2-4B	40.9	15.6	-25.3	54.6	25.6	-29.0	49.6	22.6	-27.0	16.5	0.0	-16.5	31.5	7.7	-23.8
Gemma-3-12B-IT	61.7	29.1	-32.6	65.4	33.3	-32.1	60.1	26.3	-33.8	34.3	8.7	-25.6	65.6	17.3	-48.3
Gemma-3-4B-IT	46.2	14.1	-32.1	44.3	12.8	-31.5	43.4	8.8	-34.6	16.1	0.0	-16.1	36.5	5.8	-30.7
SEA-LION-v4-8B-VL	68.2	32.2	-36.0	75.6	44.6	-31.0	83.4	49.6	-33.8	51.7	4.3	-47.4	71.7	36.5	-35.2
SEA-LION-v4-4B-VL	53.0	12.1	-40.9	53.6	16.9	-36.7	63.6	17.5	-46.1	26.5	0.0	-26.5	44.2	11.5	-32.7

Table 2: Reasoning step performance. Subjects are abbreviated as follows: Math = Mathematics, Phys = Physics, Chem = Chemistry, Bio = Biology, CS = Computer Science. The Pearson correlation between $\mathcal{P}$ and $\mathcal{R}$ is $r = 0.959$.

Subject	GPT-5.4				InternVL3.5-8B				Qwen3-VL-8B				Molmo2-8B				Gemma-3-4B-IT			
	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{F}_{wst}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{F}_{wst}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{F}_{wst}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{F}_{wst}$	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$	$\mathcal{F}_{wst}$
Math	91.1	87.8	88.0	67.8	67.7	69.8	66.2	39.9	64.2	65.1	62.7	30.1	57.6	53.6	51.5	23.6	52.3	58.9	51.2	20.1
Phys	95.0	91.7	92.2	79.2	78.7	79.5	76.6	53.9	77.6	78.2	75.9	51.7	72.8	73.0	70.2	44.0	53.7	61.0	51.4	21.4
Chem	94.3	94.1	93.4	79.0	81.6	84.1	80.7	52.5	83.7	86.1	83.5	53.4	76.3	73.5	71.9	39.4	59.3	65.0	57.6	20.7
Bio	81.8	82.1	81.1	60.3	51.0	54.0	49.5	24.0	60.3	61.7	57.9	22.9	51.4	47.9	47.0	20.6	30.1	36.3	28.6	7.7
CS	94.3	88.2	89.4	75.1	68.7	61.4	61.3	32.4	67.9	64.4	63.4	33.2	53.9	42.0	42.6	15.5	48.9	41.2	39.6	14.1

Additionally, to evaluate step-level reasoning quality, we decompose performance into **Recall** ($\mathcal{R}$), which measures whether a model covers all necessary reasoning steps, and **Precision** ($\mathcal{P}$), which measures whether model-generated steps are valid, i.e., correspond to correct reasoning rather than redundant, hallucinated, or incorrect ones, inspired by Xie et al. (2026) and Gull et al. (2026). To automate this process, we employ GPT-5.4 as an LLM judge $\mathcal{J}$ (the full prompt template is provided in Appendix C.2) to provide binary step-level decisions by comparing the model's reasoning against a ground-truth reference. For each problem instance i, let the reference solution be $\mathcal{S}^{(i)}_{\text{gold}} = \{g^{(i)}_1, \ldots, g^{(i)}_m\}$ and the model output be $\mathcal{S}^{(i)}_{\text{model}} = \{s^{(i)}_1, \ldots, s^{(i)}_n\}$. The judge's evaluations are defined as follows:

$$\mathcal{J}_{\text{rec}}(g, \mathcal{S}^{(i)}_{\text{model}}) = \begin{cases} 1 & \text{if } g \text{ is covered by } \mathcal{S}^{(i)}_{\text{model}} \\ 0 & \text{otherwise} \end{cases}, \qquad \mathcal{J}_{\text{pre}}(s, \mathcal{S}^{(i)}_{\text{gold}}) = \begin{cases} 1 & \text{if } s \text{ is valid w.r.t } \mathcal{S}^{(i)}_{\text{gold}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Instance-level metrics for a problem i are then calculated as:

$$\mathcal{R}_i = \frac{1}{m}\sum_{k=1}^{m} \mathcal{J}_{\text{rec}}(g^{(i)}_k, \mathcal{S}^{(i)}_{\text{model}}), \quad \mathcal{P}_i = \frac{1}{n}\sum_{l=1}^{n} \mathcal{J}_{\text{pre}}(s^{(i)}_l, \mathcal{S}^{(i)}_{\text{gold}}), \quad \mathcal{F}_i = \frac{2 \cdot \mathcal{P}_i \cdot \mathcal{R}_i}{\mathcal{P}_i + \mathcal{R}_i + \epsilon} \tag{3}$$

To evaluate consistency across M variants of a problem, we report the **Worst-case Step F1**:

$$\mathcal{F}_{wst} = \frac{1}{N}\sum_{i=1}^{N} \min_{j \in [1,M]} \mathcal{F}_{i,j} \tag{4}$$

4.2 Main Result

English-only Result Table 1 presents the experimental results for the English subset of Sci-ρ . Among all evaluated models, GPT-5.4 achieves the highest performance, securing over 90% average

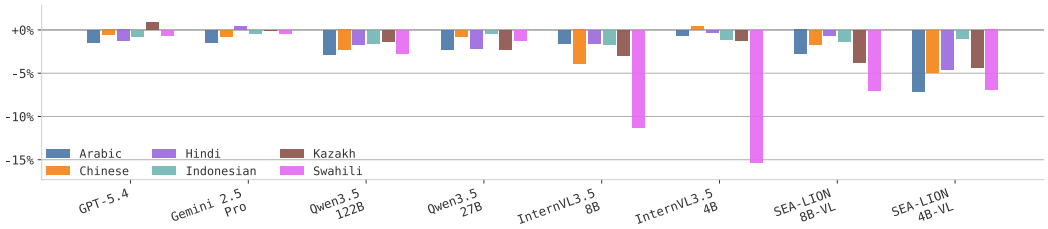


Figure 4: Average-case accuracy gap ($\Delta\mathcal{A}_{avg}$) relative to English across languages and models. Negative values indicate performance degradation in non-English settings.

accuracy in Mathematics, Physics, Chemistry, and Computer Science, with a minimum of 72.2% in Biology. Among the open-weight models, Qwen3.5-27B yields the best results at 65-90%, almost comparable to Gemini-2.5-Pro. Interestingly, Qwen3.5-27B outperforms Qwen3.5-122B-A10B in almost all subjects, likely due to the latter's Mixture-of-Experts (MoE) architecture, which only activates 10B parameters during inference. In almost every subject, SEA-LION-v4-8B-VL is the strongest model among 8-14B variants, securing an average accuracy of 51.7% in Biology and 68-83% in other subjects. Among 4B models, InternVL3.5-4B ranks highest, with 23.9% average accuracy in Biology and 54-72% in others. These results highlight the significant difficulty of the Biology subset compared to other subjects. With the exception of InternVL3.5-14B, model performance within the same family consistently scales with size.

Despite remarkable average performance across the dataset, frontier models struggle to solve problem variants consistently, exhibiting performance drops (Δ) ranging from -10% to -28% in closed-weight models. Smaller open-weight models suffer even more severe degradation. For instance, Qwen3.5-122B drops sharply in Chemistry ($\Delta = -44.9$) and Biology ($\Delta = -41.7$). SEA-LION-v4-8B-VL nearly collapses in Biology (51.7% $\rightarrow$ 4.3%, $\Delta = -47.4$), while Gemma-3-12B-IT suffers a steep decline in Computer Science (65.6% $\rightarrow$ 17.3%, $\Delta = -48.3$). Notably, under worst-case conditions, several smaller models such as InternVL3.5-4B and Molmo2-4B break down entirely in Biology, with $\Delta = -23.9$ and $\Delta = -16.5$, respectively. Furthermore, despite achieving similar average accuracy, Qwen3.5-122B-A10B has a much larger Δ across all subjects compared to Qwen3.5-27B, resulting in a substantially lower worst-case accuracy. These findings expose the inconsistency of current VLMs across problem variants, underscoring the critical need to train more robust models capable of reliable reasoning in diverse scenarios.

Multilingual Results Figure 4 compares VLM performance across seven languages by reporting $\Delta\mathcal{A}_{avg}$ relative to English. Overall, language variants have minimal effect on proprietary and large open-weight models, with GPT-5.4 and Gemini 2.5 showing near-zero degradation across all tested languages. In contrast, smaller open-weight models exhibit notable performance drops in non-English settings, with degradation becoming more pronounced in lower-resource languages. For example, Swahili exhibits the most substantial accuracy gap across smaller models, reflecting its underrepresentation in pretraining corpora. By comparison, higher-resource languages such as Chinese and Indonesian demonstrate greater resilience, showing only mild degradation.

Reasoning Step Performance To analyze reasoning quality of the models, we conduct step-level evaluation. Due to constrained budget, we selected a representative subset of five models covering a broad range of performance levels reported in Table 1. Observed from Table 2, across all evaluated models and subjects, $\mathcal{P}$ and $\mathcal{R}$ exhibit a very high correlation, suggesting that reasoning quality operates as a general capability rather than a precision–recall tradeoff, with both metrics largely reflecting the model's underlying reasoning correctness rather than separate tendencies to over- or under-generate steps.

A consistent gap between $\mathcal{F}$ and $\mathcal{F}_{\text{wst}}$ across all evaluated models reveals reasoning step fragility, most pronounced in Mathematics and Computer Science. Notably, even GPT-5.4 drops from $\mathcal{F}$ to $\mathcal{F}_{\text{wst}}$ in the range of 13.0 to 20.2 points across subjects, suggesting that worst-case reasoning fragility is a general phenomenon rather than a weakness of smaller models alone. This degradation worsens for weaker models, Gemma-3-4B-IT degrades from $\mathcal{F}$=39.6 to $\mathcal{F}_{\text{wst}}$=14.1 in Computer Science and from $\mathcal{F}$=28.6 to $\mathcal{F}_{\text{wst}}$=7.7 in Biology, suggesting that average F1 alone may overestimate true

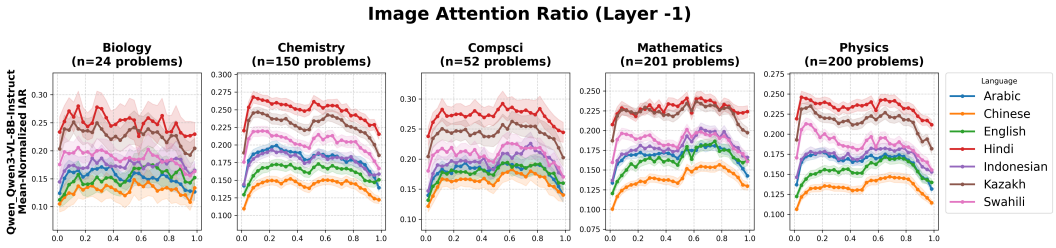


Figure 5: IAR trends across subjects and languages for *Qwen3-VL-8B-Instruct* (95% confidence intervals are shown).

reasoning capability. At the subject level, Mathematics exhibits the largest $\mathcal{F}$-to-$\mathcal{F}_{\text{wst}}$ drop across nearly all models, while Chemistry and Physics consistently yield stronger step-level performance. Biology remains the most challenging across all evaluated models, aligning with the accuracy trends in Table 1.

4.3 Attention Heads Analysis

To gain deeper insights into the model's internal reasoning during response generation, we examined the *Qwen3-VL-8B-Instruct*'s attention heads at the final layer. Inspired by previous study on VLMs modality bias (Chen et al., 2025; Deng et al., 2025; Zheng et al., 2025) and attention heads inspection during output generation (Yin et al., 2025), this experiment quantifies the extent to which the model attends to visual tokens relative to textual tokens.

When generating token t, we first average the attention weights of all previous context tokens across all H heads. Then, we calculate the average attention directed towards the image tokens (I) and the average attention directed towards the text tokens (T):

$$A_I = \frac{1}{|I|} \sum_{i \in I} \bar{a}_{t,i} \quad \text{and} \quad A_T = \frac{1}{|T|} \sum_{k \in T} \bar{a}_{t,k}$$

where $\bar{a}_{t,i}$ and $\bar{a}_{t,k}$ represent the head-averaged attention weights from the generated token t to image token i and text token k, respectively, and $|I|$ and $|T|$ denote the total number of tokens within the image and text sets. Finally, we calculate the **Image Attention Ratio** (IAR) by dividing the average image attention by the sum of both:

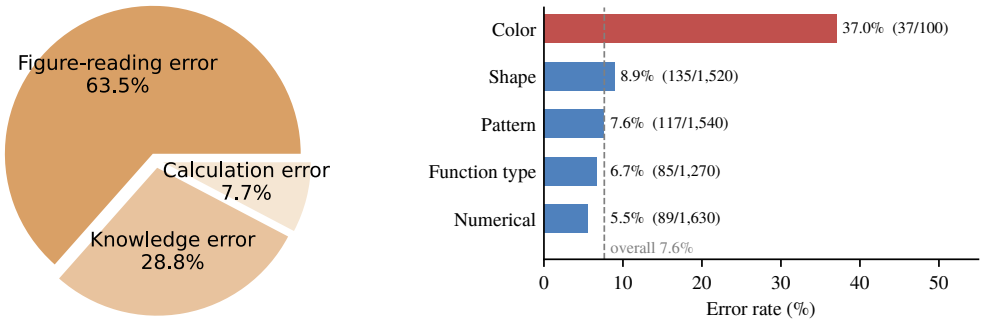
$$\text{IAR} = \frac{A_I}{A_I + A_T}$$

In our experiment, we excluded special tokens (e.g., the start token), the prompt template, and newly generated response tokens to ensure only tokens related to the problem were measured. We calculated the IAR for the entire response to measure the trend of image-component utilization as text generation proceeds. We then normalized the length of all responses to a scale of 0 to 1. Finally, we averaged the results across all problems within the same subject.

Figure 5 illustrates the IAR trends across the five subjects. The results demonstrate that the IAR varies across languages, with Chinese exhibiting the lowest ratio. This indicates that the model relies less on the visual context when prompted in Chinese. In contrast, queries in Kazakh and Hindi yield the highest IARs, indicating a much stronger reliance on the image. These findings highlight a difference in how the model balances text and image information across languages. We hypothesize that higher language familiarity allows the model to rely more on text part of the inputs, whereas lower familiarity forces it to compensate by depending more heavily on visual inputs.

4.4 Additional Analysis

Exact Computation vs Visual Intuition We observed a notable failure case where the frontier model, GPT-5.4, preferred to rely on visual intuition rather than computing the exact answer, as illustrated in Figure 1. In problem $Mat020$ (variant 04), the model successfully computed the answer



(a) Distribution of error types.

(b) Error rate by perturbation type.

Figure 6: Error analysis of GPT-5.4 on the English subset. (a) Distribution of error types across 52 manually inspected templates. (b) Error rate for each perturbation type, with incorrect answers over total instances in parentheses. The dashed line marks the overall error rate.

based on the provided numerical information. However, in variant 07, the model initially follows the correct reasoning path to derive the exact answer, but is then distracted by the accompanying figure, which visually depicts point T located close to $x = 0$. This ultimately leads to an incorrect final answer. This behavior highlights that, at boundary conditions, visual cues can override a model's logical reasoning and distract it from computing the exact solution.

Failure Mode Analysis To better understand how VLMs fail to answer our problems, we conducted a manual inspection of 52 templates where GPT-5.4 produced incorrect answers. We categorized the failures into three common error types: (1) **figure-reading errors** occur when the VLM fails to extract accurate numerical values or miscounts number of objects within the image (2) **knowledge errors** happen when the model demonstrates flawed logical reasoning, such as using inapplicable theorems, declaring unstated assumptions, or hallucinating non-existent information (3) **calculation errors** occur when the VLM fails to perform numerical computations correctly, despite following the correct underlying logic and steps. Figure 6a illustrates the distribution of these errors. Figure-reading errors are the most frequent, accounting for 63.5%, followed by knowledge errors at 28.8%, with calculation errors comprising the remaining 7.7%. These findings emphasize that future advancements in VLM development must prioritize robust architectural designs and training methodologies that enhance precise visual grounding and mitigate hallucinations.

Error Breakdown by Perturbation Type To understand which variations drive the gap between average-case and worst-case accuracy, we computed the error rate of GPT-5.4 for each perturbation type on the English subset (Figure 6b). Color perturbations yield the highest error rate at 37.0%, although only 100 instances feature Color variations. All other types stay below 10%, with Numerical variations lowest at 5.5%. Shape and Pattern variations together account for 252 of the 463 incorrect answers (54.4%), in line with figure-reading errors being the most frequent error type (Figure 6a).

5 Conclusion

Our work introduce Sci-ρ as a visually-grounded symbolic benchmark in five STEM subjects and seven languages. We collect responses from 17 VLMs and evaluate their performance using six metrics. Our experiments reveal a noticeable performance gap between average-case and worst-case accuracy across all models and subjects, while the effect of multilinguality is only apparent in small open-weight models. Similar performance trends are also observed when the models are evaluated based on their reasoning steps. Finally, we discover cross-lingual variation in the relative attention allocated to image tokens compared to text tokens when inspecting the attention heads of *Qwen3-VL-8B-Instruct*. Our work highlights the importance of evaluation beyond static benchmarks for measuring the quality of VLMs.

AI USE STATEMENT

In this work, we used generative AI tools for generate synthetic datasets, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean and reformat dataset, support qualitative and thematic data analysis. We have not used generative AI tools for interpret results, and "help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses" are not applicable to this work. Additionally, we used generative AI tools for create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, summarize or analyze existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, edit a research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work. The AI-translated dataset is verified by native speakers of corresponding languages. We only used AI to find prior works and manually draft the related works section to prevent the possibility of AI hallucination. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

AI Singapore. Sea-lion v4 model documentation. https://docs.sea-lion.ai/models/sea-lion-v4, 2026. Accessed: 2026-04-26.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Shuo Chen, Jianzhe Liu, Zhen Han, Yan Xia, Daniel Cremers, Philip Torr, Volker Tresp, and Jindong Gu. True multimodal in-context learning needs attention to the visual context, 2025. URL https://arxiv.org/abs/2507.15807.

Christopher Clark, Jieyu Zhang, Zixian Ma, Jae Sung Park, Mohammadreza Salehi, Rohun Tripathi, Sangho Lee, Zhongzheng Ren, Chris Dongjoo Kim, Yinuo Yang, Vincent Shao, Yue Yang, Weikai Huang, Ziqi Gao, Taira Anderson, Jianrui Zhang, Jitesh Jain, George Stoica, Winson Han, Ali Farhadi, and Ranjay Krishna. Molmo2: Open weights and data for vision-language models with video understanding and grounding, 2026. URL https://arxiv.org/abs/2601.10611.

Ailin Deng, Tri Cao, Zhirui Chen, and Bryan Hooi. Words or vision: Do vision-language models have blind faith in text?, 2025. URL https://arxiv.org/abs/2503.02199.

Google DeepMind. Gemini 2.5 flash model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf, 2025a. Accessed: 2026-04-26.

Google DeepMind. Gemini 2.5 pro model card. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Pro-Model-Card.pdf, 2025b. Accessed: 2026-04-26.

Ayesha Gull, Muhammad Usman Safder, Rania Elbadry, Fan Zhang, Veselin Stoyanov, Preslav Nakov, and Zhuohan Xie. Engtrace: A symbolic benchmark for verifiable process supervision of engineering reasoning, 2026. URL https://arxiv.org/abs/2511.01650.

Ziyu Guo, Ray Zhang, Hao Chen, Jialin Gao, Dongzhi Jiang, Jiaze Wang, and Pheng-Ann Heng. Sciverse: Unveiling the knowledge comprehension and visual reasoning of lmms on multi-modal scientific problems, 2025. URL https://arxiv.org/abs/2503.10627.

Patrick Amadeus Irawan, Ikhlasul Akmal Hanif, Muhammad Dehan Al Kautsar, Genta Indra Winata, Fajri Koto, and Alham Fikri Aji. Vision language models are confused tourists. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 1763–1773, June 2026.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering, 2022. URL https://arxiv.org/abs/2209.09513.

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, 2024. URL https://arxiv.org/abs/2310.02255.

Meta AI. Introducing llama 4, 2025. URL https://ai.meta.com/llama/. Accessed: 2026-04-16.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2025. URL https://arxiv.org/abs/2410.05229.

Shravan Nayak, Kanishk Jain, Rabiul Awal, Siva Reddy, Sjoerd Van Steenkiste, Lisa Anne Hendricks, Karolina Stanczak, and Aishwarya Agrawal. Benchmarking vision language models for cultural understanding. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5769–5790, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.329. URL https://aclanthology.org/2024.emnlp-main.329/.

Jean De Dieu Nyandwi, Yueqi Song, Simran Khanuja, and Graham Neubig. Grounding multilingual multimodal LLMs with cultural knowledge. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 24198–24242, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1232. URL https://aclanthology.org/2025.emnlp-main.1232/.

Shi Qiu, Shaoyang Guo, Zhuo-Yang Song, Yunbo Sun, Zeyu Cai, Jiashen Wei, Tianyu Luo, Yixuan Yin, Haoxu Zhang, Yi Hu, Chenyang Wang, Chencheng Tang, Haoling Chang, Qi Liu, Ziheng Zhou, Tianyu Zhang, Jingtian Zhang, Zhangyi Liu, Minghao Li, Yuku Zhang, Boxuan Jing, Xianqi Yin, Yutong Ren, Zizhuo Fu, Jiaming Ji, Weike Wang, Xudong Tian, Anqi Lv, Laifu Man, Jianxiang Li, Feiyu Tao, Qihua Sun, Zhou Liang, Yushu Mu, Zhongxuan Li, Jing-Jun Zhang, Shutao Zhang, Xiaotian Li, Xingqi Xia, Jiawei Lin, Zheyu Shen, Jiahang Chen, Qiuhao Xiong, Binran Wang, Fengyuan Wang, Ziyang Ni, Bohan Zhang, Fan Cui, Changkun Shao, Qing-Hong Cao, Ming xing Luo, Yaodong Yang, Muhan Zhang, and Hua Xing Zhu. Phybench: Holistic evaluation of physical perception and reasoning in large language models, 2025. URL https://arxiv.org/abs/2504.16074.

Yufan Ren, Konstantinos Tertikas, Shalini Maiti, Junlin Han, Tong Zhang, Sabine Süsstrunk, and Filippos Kokkinos. Vgrp-bench: Visual grid reasoning puzzle benchmark for large vision-language models, 2025. URL https://arxiv.org/abs/2503.23064.

Risa Shinoda, Nakamasa Inoue, Hirokatsu Kataoka, Masaki Onishi, and Yoshitaka Ushiku. Agrobench: Vision-language model benchmark in agriculture, 2025. URL https://arxiv.org/abs/2507.20519.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra,

Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Pe-

terson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

Kai Sun, Yushi Bai, Ji Qi, Lei Hou, and Juanzi Li. MM-MATH: Advancing multimodal math evaluation with process evaluation and fine-grained classification. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 1358–1375, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.73. URL https://aclanthology.org/2024.findings-emnlp.73/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

An Vo, Khai-Nguyen Nguyen, Mohammad Reza Taesiri, Vy Tuong Dang, Anh Totti Nguyen, and Daeyoung Kim. Vision language models are biased, 2026. URL https://arxiv.org/abs/2505.23941.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset, 2024. URL https://arxiv.org/abs/2402.14804.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, Jingjing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025. URL https://arxiv.org/abs/2508.18265.

Kun Xiang, Heng Li, Terry Jingchen Zhang, Yinya Huang, Zirong Liu, Peixin Qu, Jixi He, Jiaqi Chen, Yu-Jie Yuan, Jianhua Han, Hang Xu, Hanhui Li, Mrinmaya Sachan, and Xiaodan Liang. Seephys: Does seeing help thinking? – benchmarking vision-based physics reasoning, 2025. URL https://arxiv.org/abs/2505.19099.

Zhuohan Xie, Daniil Orel, Rushil Thareja, Dhruv Sahnan, Hachem Madmoun, Fan Zhang, Debopriyo Banerjee, Georgi Georgiev, Xueqing Peng, Lingfei Qian, Jimin Huang, Jinyan Su, Aaryamonvikram Singh, Rui Xing, Rania Elbadry, Chen Xu, Haonan Li, Fajri Koto, Ivan Koychev, Tanmoy Chakraborty, Yuxia Wang, Salem Lahlou, Veselin Stoyanov, Sophia Ananiadou, and Preslav Nakov. Finchain: A symbolic benchmark for verifiable chain-of-thought financial reasoning, 2026. URL https://arxiv.org/abs/2506.02515.

Qingyu Yin, Chak Tou Leong, Linyi Yang, Wenxuan Huang, Wenjie Li, Xiting Wang, Jaehong Yoon, YunXing, XingYu, and Jinjin Gu. Refusal falls off a cliff: How safety alignment fails in reasoning?, 2025. URL https://arxiv.org/abs/2510.06036.

Xinyu Zhang, Yuxuan Dong, Yanrui Wu, Jiaxing Huang, Chengyou Jia, Basura Fernando, Mike Zheng Shou, Lingling Zhang, and Jun Liu. Physreason: A comprehensive benchmark towards physics-based reasoning, 2025. URL https://arxiv.org/abs/2502.12054.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors, 2024. URL https://arxiv.org/abs/2309.03882.

Xu Zheng, Chenfei Liao, Yuqian Fu, Kaiyu Lei, Yuanhuiyi Lyu, Lutao Jiang, Bin Ren, Jialei Chen, Jiawen Wang, Chengxin Li, Linfeng Zhang, Danda Pani Paudel, Xuanjing Huang, Yu-Gang Jiang, Nicu Sebe, Dacheng Tao, Luc Van Gool, and Xuming Hu. Mllms are deeply affected by modality bias, 2025. URL https://arxiv.org/abs/2505.18657.

Yongshuo Zong, Tingyang Yu, Ruchika Chavhan, Bingchen Zhao, and Timothy Hospedales. Fool your (vision and) language model with embarrassingly simple permutations, 2024. URL https://arxiv.org/abs/2310.01651.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models, 2025. URL https://arxiv.org/abs/2411.00836.

A Additional Information for Dataset Construction

A.1 Curriculum and Taxonomy

Cambridge IGCSE and A-Level The Cambridge International General Certificate of Secondary Education (IGCSE) and the Cambridge Advanced Level (A-Level) are international secondary education qualifications developed by Cambridge Assessment International Education. IGCSE is a foundational secondary curriculum typically taken by students aged 14–16, while A-Level is an advanced pre-university curriculum for students aged 16–18. Both curricula define a standardized scope of high school STEM competencies, which we adopt as the basis of our topic hierarchy.

Vision-Dependent Materials We define vision-dependent materials as subtopics in which text alone is insufficient to describe or solve a problem, so that the image carries information required to reach the answer. Examples include electric circuit analysis, graph interpretation (e.g., finding the tangent of a parabola), dimensions of geometric shapes, and molecular structures. Subtopics such as pure algebraic equation solving or text-only word problems are less vision-dependent, and we did not include them in our dataset.

Topics and Subtopics Our taxonomy has two levels. A topic contains one or more subtopics, and each subtopic belongs to exactly one topic. Each template is assigned to exactly one subtopic, and therefore to exactly one topic.

A.2 Template Creation

The dataset was constructed through a collaboration between the authors and hired contributors. All individuals involved were proficient in Python and the corresponding STEM subject, with backgrounds as STEM graduates, advanced undergraduates, or science olympiad medalists. Their responsibilities included understanding the vision-dependent subtopics of the corresponding subject, developing dynamic problem templates (complete with images, descriptions, solutions, and reasoning steps), and cross-validating each other's work. The topics and subtopics were organized beforehand by the authors, predominantly based on the Cambridge IGCSE and A-Level curricula (Appendix A.1). In total, eight people (including authors and non-authors) participated in the construction of the 606 templates. All hired contributors (non-authors) were compensated at a rate above the minimum wage of their region of residence.

The template creation process began with a one-hour workshop, in which we explained the research objectives and workflows, including a live demonstration of creating a template. Following this, the contributors were given five weeks (25 working days) to develop the code, with a maximum limit of 100 templates per person. Table 3 summarizes the written guideline given to the contributors. Some details shown during the workshop demonstration may not be fully captured in the written guideline. Contributors could adapt problems from existing sources, create entirely new problems, or combine both. When adapting a problem, they were required to record its source in the dataset.

Table 3: Guideline given to contributors for template creation.

Guideline – Template Creation

A **template** is the unit of problem in the dataset, and is represented by a Python class.

Steps:

1. Open the syllabus of the corresponding subject and curriculum.
2. Read and understand the competency standards of a subtopic, including its scope and limitations.
3. Write the template problem for the subtopic. Make sure the problem fits the subtopic requirements.
4. Write the solution steps in LaTeX with the help of an LLM (e.g., GPT-5 or Gemini), using the provided prompt template (Table 5).
5. Run `python generate_dataset.py --subject physics --debug True` to generate the instances.
6. Open `dataset/physics/dataset.json` and the corresponding images to verify that the instances were generated successfully.
7. Open `dataset/physics/debug/[file].txt` to check that the LaTeX solution is written correctly (e.g., by rendering it with QuickLaTeX).

Note: You do not have to come up with question ideas from scratch. You may search for questions on similar materials online (e.g., previous benchmarks) and use them as inspiration. However, every question must have a clear source, recorded in the `source` field of the dataset.

Variable Sampling All randomly generated variables in a template (e.g., `self.time`) are instantiated with Python's `random` module using a predetermined seed, which makes every instance

reproducible. The sampling function depends on the needs of the problem: `random.randint` for integers, `random.choice` for selecting from a list of candidates (e.g., function types or colors), and `random.uniform` for floating-point values. We set the range of each variable to resemble the values found in typical STEM exams, and added constraints so that no sampled value is physically impossible. For instance, the number of atoms must be an integer, and the speed of an object cannot exceed the speed of light.

Image Generation Almost all images are drawn programmatically using graphics libraries such as `matplotlib` or `PIL`. When an instance is generated, the code reads the sampled variables (e.g., `self.voltage`) and draws the image from scratch, so that the text labels, curves, shapes, and object counts in the image always match the description. A small number of templates start from an existing base diagram in PNG format. For these templates, the values displayed on the original diagram are overwritten with the newly sampled values by overlaying text at fixed coordinates using `matplotlib.pyplot.text`.

Solution and Verification The contributors wrote the reasoning steps of each template in LaTeX format. They were allowed to use AI to convert the Python computation into LaTeX (the prompt template is shown in Table 5). After that, the contributors generated instances and ensured that all information was generated correctly. Contributors were required to revise the code if any component failed to generate properly. We also used an automated script to verify that each template could generate at least 10 variants, where each variant differs from the others in at least the description or the image.

For quality checking, the contributors cross-validated each other's work if at least two people worked on the same subject. When only a single contributor was assigned to a subject, a contributor from a different subject reviewed their work. Reviewers verified the correctness of each template, including the problem category, description, answer, image, and reasoning, and generated instances to make sure the template was written correctly.

A.3 Translation

The dataset was translated into six languages (Indonesian, Chinese, Arabic, Hindi, Kazakh, and Swahili) to evaluate VLM performance in non-English contexts. First, all problem descriptions from the Python templates were consolidated into a single spreadsheet. We then used Gemini 3 Pro to generate initial translation drafts, using the prompt template in Table 6. We selected this model for its proficiency in multilingual contexts and its ability to preserve Python syntax.

The translation and verification process involved a collaboration between authors and hired contributors. To ensure consistency with the original dataset, we prioritized contributions from authors and workers already involved in the English dataset development. When no native speaker of a target language was present among the primary contributors, we hired external native-speaking translators for that language. All individuals involved in the translation verification were native speakers of the target language, highly familiar with A-Level STEM terminology, and proficient in Python.

All translators were given a guideline document to ensure consistency, summarized in Table 4. The guideline asks translators to review the machine-translated draft critically against the English original, handle scientific terminology according to the conventions of the target language, and keep the Python and LaTeX syntax intact so that every translated description still runs inside its template. The verification of 606 descriptions was completed over a three-week period. To maintain quality control, we intentionally introduced errors into the dataset by deleting the middle 25% of 40 randomly chosen descriptions (approximately 7% of the total). Translators who failed to identify and correct these errors were required to revise their work, as this served as a proxy for attention to detail. Upon successful completion of the task, all annotators were compensated at competitive rates.

B Variants in Our Dataset

Figure 7 presents examples of the variants in our dataset. We identify five types of variants in our dataset: pattern, numerical, shape, color, and function type.

Table 4: Summary of the guideline given to translators for verifying machine-translated descriptions.

Guideline – Translation Verification

Your task is to review and refine AI-generated translations of STEM problems. The final text must be scientifically accurate, natural in the target language, and correctly formatted for Python code integration.

Step 1 – Compare Source and Draft

- Read the English original first to understand the discipline and context (e.g., kinematics, calculus, organic chemistry, or genetics).
- Do not assume the AI draft is correct. Look for awkward phrasing, literal translations of idioms, and misunderstandings of the underlying science.

Step 2 – Scientific Terminology

- Follow the educational standard of the target language for acronyms (e.g., DNA becomes ADN in Spanish). Keep English letters if the local curriculum does.
- Use established localized terms where they exist (e.g., "velocity" becomes "kecepatan" in Indonesian). Keep a term in English if no widely used equivalent exists.
- Never translate or transliterate Latin binomial names (e.g., *Escherichia coli*). Translate the common name if given.
- Translate IUPAC chemical names (e.g., sodium chloride) into the formal chemical nomenclature of the target language.
- Preserve the letters of variables and constants (e.g., do not change F for Force into K for *Kraft* in German).

Step 3 – Protect Python Syntax

- Do not translate, re-case, or remove anything inside `{ }`. The block may be moved within the sentence if grammar requires it.
- Keep double backslashes and escape characters (e.g., `\\frac`, `\n`) exactly as in the original.
- Respect string boundaries. Escape any quote character that matches the outer wrapper.
- Do not insert hard line breaks unless the original uses triple quotes or explicit `\n`.
- Localize decimal separators in plain text only. Never change format specifiers such as `{self.speed:.2f}`.
- Keep string prefixes (`f`, `r`, `rf`) as they are.

Step 4 – Mathematical Symbols and Units

- Keep scientific and mathematical notation exactly as it is.
- For right-to-left languages (e.g., Arabic), make sure formulas and chemical reactions still read left to right.
- Preserve the `$` math delimiters. Do not place translated words inside `$...$` or `\text{}`, since the math font used for rendering cannot display non-Latin characters.

Step 5 – Natural Phrasing

- Rewrite robotic sentences so they read naturally to a native speaker, in the formal register of a STEM textbook or exam.
- Use the standard localized form of units where that is the convention in the target language.

Additional Notes

- Translate references to visual elements (e.g., "the shaded region", "point A") literally, without re-wording elements you cannot see.
- Only modify the column containing the translated text.

Table 5: Prompt template for converting Python reasoning to LaTeX format.

Prompt Template – Generating LaTeX

As an expert in Python and LaTeX formatting, your task is to convert the reasoning process of my Python code into a LaTeX-formatted explanation. Store the result in a single variable named `self.reasoning`.

Guidelines:

- Use f-strings in Python format (with double backslashes for LaTeX newlines, e.g. `\\`).
- Wrap the entire reasoning inside `\begin{aligned} ... \end{aligned}` and use `\\` for line breaks between aligned equations.
- Each mathematical step must appear on a new line using `\\`.
- Use clean, step-by-step mathematical reasoning with proper LaTeX formatting.
- Include intermediate symbolic steps before substituting numeric values.
- Include units if applicable (e.g. kg, m, s).
- Only output the final Python code (no extra text, markdown, or explanations). Do not write anything else.

EXAMPLE INPUT

```
self.description = "A box experiences two perpendicular forces as
shown in the diagram.  The magnitude of the resultant force is (_) N"
self.x_direction = rnd.randint(-18, 18)
self.y_direction = rnd.randint(-18, 18)
while self.x_direction in range(-5, 6) or self.y_direction in
range(-5, 6):
    self.x_direction = rnd.randint(-18, 18)
    self.y_direction = rnd.randint(-18, 18)
self.resultant = math.sqrt(self.x_direction**2 + self.y_direction**2)
self.answer = round(self.resultant, 2)
```

EXPECTED OUTPUT

```
self.reasoning = (
    "\[\begin{aligned}"
    f"\text{{Horizontal component}} &= {self.x_direction}\,\text{{N}}\\"
    f"\text{{Vertical component}} &= {self.y_direction}\,\text{{N}}\\"
    f"\text{{Resultant force is given by:}} &\\"
    f"R &= \sqrt{{x^2 + y^2}}\\"
    f"&= \sqrt{{({self.x_direction})^2 + ({self.y_direction})^2}}\\"
    f"&= \sqrt{{{self.x_direction**2} + {self.y_direction**2}}}\\"
    f"&= {self.answer}\,\text{{N}}"
    "\end{aligned}\]"
)
```

Now, here is the actual Python code you need to explain:

```
[CODE]
```

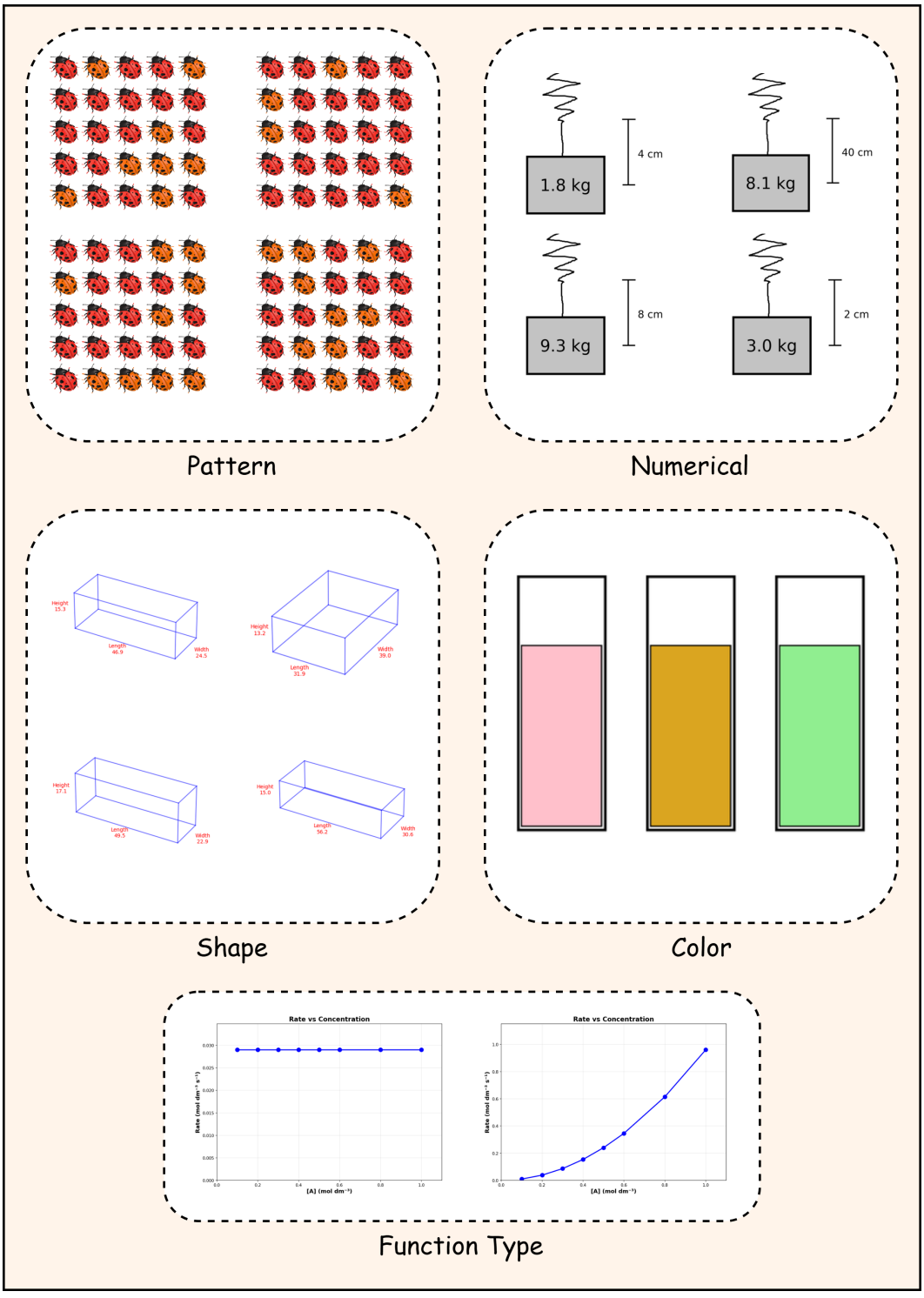


Figure 7: More examples of variants in our dataset.

Table 6: Prompt template for translation. During the experiment, we prompted Gemini 3 Pro with a batch of 100 line of codes per prompt.

Prompt Template – Translation

Translate the following Python f-strings into [LANGUAGE].

Please strictly follow these rules:

1. **Format:** Output the translated code vertically, with exactly one string assignment per line so it is easy to copy-paste. Do not add extra line breaks inside the strings.
2. **Variable Name:** Use the exact variable name `self.description_[LANGUAGE]` for every line.
3. **Code Syntax:** Do NOT translate, change the casing, or alter any variables inside the curly brackets `{ }` (e.g., `{self.sample_size}`).
4. **STEM Accuracy:** Maintain formal scientific and mathematical accuracy. Keep universal abbreviations (DNA, ATP, etc.) in English.
5. **String Formatting & Escaping:** Mirror the exact string prefix (`f"..."` or `"..."`), the exact backslash escaping (`\frac` vs `\\frac`), and other formatting from the original.

Code to translate:
`[CODE]`

C Experiment Setup

C.1 Model Artifacts

Figure 7 displays the model parameters used for the experiment. By default, we used cloud-based GPU to collect model responses of small open-weight models. For large open-weight models ($\geq$27B), we used API-based inference for convenience.

C.2 Prompt Template

We provide the prompt templates used in our experiments. Table 6 presents the translation prompt. Table 5 shows the template for converting code steps into LaTeX steps. Table 8 contains the instruction prompt for evaluation. Finally, the prompt templates used by the judge model for step-level reasoning evaluation (precision and recall) are given in Table 9 and 10.

Table 7: Model parameters used in the experiments.

Model	Parameters
GPT-5.4	model_name=gpt-5.4-2026-03-05 temperature=1.0 max_completion_tokens=4096 reasoning_effort="none"
Gemini-2.5-Pro	model_name=gemini-2.5-pro temperature=0.0 max_output_tokens=8192 thinkingBudget=2048
Gemini-2.5-Flash	model_name=gemini-2.5-flash temperature=0.0 max_output_tokens=4096 thinkingBudget=0
Qwen3.5-122B-A10B	model_name=Qwen/Qwen3.5-122B-A10B temperature=1.0 max_completion_tokens=8192 reasoning=False
Qwen3.5-27B	model_name=Qwen/Qwen3.5-27B temperature=1.0 max_completion_tokens=8192 reasoning=False
Llama-4-Scout	model_name=meta-llama/Llama-4-Scout-17B-16E-Instruct temperature=0.0 max_completion_tokens=4096
InternVL3.5-14B	model_name=OpenGVLab/InternVL3_5-14B temperature=0.0 max_tokens=4096
InternVL3.5-8B	model_name=OpenGVLab/InternVL3_5-8B temperature=0.0 max_tokens=4096
InternVL3.5-4B	model_name=OpenGVLab/InternVL3_5-4B temperature=0.0 max_tokens=4096
Qwen3-VL-8B-Instruct	model_name=Qwen/Qwen3-VL-8B-Instruct temperature=0.0 max_tokens=4096
Qwen3-VL-4B-Instruct	model_name=Qwen/Qwen3-VL-4B-Instruct temperature=0.0 max_tokens=4096
Molmo2-8B	model_name=allenai/Molmo2-8B temperature=0.0 max_tokens=4096
Molmo2-4B	model_name=allenai/Molmo2-4B temperature=0.0 max_tokens=4096
Gemma-3-12B-IT	model_name=google/gemma-3-12b-it temperature=0.0 max_tokens=4096
Gemma-3-4B-IT	model_name=google/gemma-3-4b-it temperature=0.0 max_tokens=4096
SEA-LION-v4-8B-VL	model_name=aisingapore/Qwen-SEA-LION-v4-8B-VL temperature=0.0 max_tokens=4096
SEA-LION-v4-4B-VL	model_name=aisingapore/Qwen-SEA-LION-v4-4B-VL temperature=0.0 max_tokens=4096

Table 8: Prompt template for response generation.

Prompt Template – Response Generation

Solve the above problem step-by-step. You must write your mathematical or logical reasoning strictly in English within the 'solution' field. Format your reasoning neatly into explicitly numbered steps separated by newlines (e.g., "Step 1: ...\nStep 2: ..."). Then, extract your final answer into the 'short answer' field. Your response must be valid JSON and strictly adhere to the following format. Do not include any text outside of the JSON block.

```
{
    "solution": "[Step-by-step solution]",
    "short answer": "[Final numerical value only. Do not include
units]"
}
```

Table 9: Prompt template for the recall judge ($\mathcal{J}_{\text{rec}}$).

Prompt Template – Recall Judge

You are a STEM reasoning evaluator.

GROUND TRUTH SOLUTION: `"""{gold_solution}"""`

MODEL OUTPUT: `"""{model_output}"""`

PROBLEM DESCRIPTION: `"""{problem_description}"""`

If a figure is attached, use it as additional context for judging the reasoning. Both texts contain steps labeled "Step 1", "Step 2", etc. Extract them regardless of formatting (colons, bold, newlines, indentation may vary).

First, count the number of steps in the **GROUND TRUTH**. Call this m. For **EVERY** gold step 1 to m, decide if the model output covers it. You **MUST** include every integer from 1 to m as a key — no skipping.

A gold step is **COVERED** if the model addresses the same operation with equivalent reasoning. Accept different notation or variable names, but **NOT** wrong values or wrong operations.

Respond **ONLY** with this JSON, no other text:

```
{
    "m": <number of gold steps>,
    "coverage": {"1": true, "2": false, ..., "<m>": true},
    "rationale": {"1": "one sentence", ..., "<m>": "one sentence"}
}
```

Table 10: Prompt template for the precision judge ($\mathcal{J}_{\text{pre}}$).

Prompt Template – Precision Judge

You are a STEM reasoning evaluator.

GROUND TRUTH SOLUTION: `"""{gold_solution}"""`

MODEL OUTPUT: `"""{model_output}"""`

PROBLEM DESCRIPTION: `"""{problem_description}"""`

If a figure is attached, use it as additional context for judging the reasoning. Both texts contain steps labeled "Step 1", "Step 2", etc. Extract them regardless of formatting (colons, bold, newlines, indentation may vary).

First, count the number of steps in the **MODEL OUTPUT**. Call this n. For **EVERY** model step 1 to n, decide if it is valid given the ground truth. You **MUST** include every integer from 1 to n as a key — no skipping.

A model step is **VALID** if any of these are true: (1) it matches a ground truth step directly; (2) it is a correct sub-step of a ground truth step (splitting is fine); (3) it is a correct intermediate that logically follows.

A model step is **INVALID** if any of these are true: (1) it restates a step already completed (padding); (2) it applies a wrong formula, wrong value, or wrong operation; (3) it introduces reasoning not grounded in the problem (hallucination).

Respond **ONLY** with this JSON, no other text:

```
{
    "n": <number of model steps>,
    "validity": {"1": true, "2": false, ..., "<n>": true},
    "rationale": {"1": "one sentence", ..., "<n>": "one sentence"}
}
```

D ADDITIONAL RESULTS

This section presents fine-grained results omitted from the primary paper. Table 11 details the average-case accuracy across all languages, while Table 12 provides the corresponding worst-case accuracy metrics.

Table 11: Average-case accuracy on Sci-ρ , over all languages

Model	ALL	English	Arabic	Chinese	Hindi	Indonesian	Kazakh	Swahili
Closed-source Models								
GPT-5.4	88.8	89.4	88.0	88.8	88.1	88.6	90.2	88.7
Gemini-2.5-Pro	85.8	86.2	84.8	85.5	86.7	85.8	86.1	85.8
Gemini-2.5-Flash	80.4	80.7	80.2	80.3	80.3	80.7	80.2	80.2
Open-source Models								
Qwen3.5-122B-A10B	80.8	82.6	79.8	80.3	81.0	81.0	81.3	79.9
Llama-4-Scout	69.2	69.6	68.1	70.0	69.8	69.4	69.0	68.3
InternVL3.5-14B	55.4	60.0	57.6	56.0	57.5	59.1	54.3	43.5
InternVL3.5-8B	59.8	63.1	61.5	59.1	61.5	61.4	60.2	51.8
InternVL3.5-4B	51.6	54.2	53.6	54.7	53.9	53.1	53.0	38.9
Qwen3-VL-8B-Instruct	65.3	67.2	66.8	67.5	62.3	66.8	66.2	60.1
Qwen3-VL-4B-Instruct	26.8	29.9	21.8	29.0	26.5	30.0	22.7	27.8
Molmo2-8B	43.9	46.4	43.9	44.5	44.4	47.2	43.7	37.5
Molmo2-4B	34.0	38.6	35.6	38.1	33.6	34.3	32.9	25.1
Gemma-3-12B-IT	53.6	57.4	51.7	54.8	52.4	51.5	54.3	53.5
Gemma-3-4B-IT	36.4	37.3	36.1	37.0	35.9	37.2	36.3	35.1
SEA-LION-v4-8B-VL	67.7	70.1	67.3	68.4	69.5	68.8	66.4	63.1
SEA-LION-v4-4B-VL	44.1	48.2	41.1	43.3	43.6	47.2	43.8	41.2
qwen-qwen3-5-27b	82.3	83.6	81.3	82.8	81.4	83.2	81.3	82.3

Table 12: Worst-case accuracy on Sci-ρ across all languages.

Model	ALL	English	Arabic	Chinese	Hindi	Indonesian	Kazakh	Swahili
Closed-source Models								
GPT-5.4	73.9	73.6	74.6	74.0	72.6	72.9	75.6	74.2
Gemini-2.5-Pro	62.5	64.8	61.6	59.8	63.7	62.1	61.9	63.4
Gemini-2.5-Flash	56.6	57.3	55.9	57.2	54.7	56.6	56.6	58.1
Open-source Models								
Qwen3.5-122B-A10B	43.6	45.1	42.6	41.1	44.5	43.2	44.1	44.4
Llama-4-Scout	33.7	31.4	34.6	36.3	36.7	32.1	32.3	32.5
InternVL3.5-14B	18.4	22.4	22.4	18.6	19.0	21.5	14.9	9.8
InternVL3.5-8B	25.8	30.1	27.4	23.7	26.8	26.4	26.2	19.8
InternVL3.5-4B	17.5	19.0	18.4	19.1	17.6	18.6	18.3	11.3
Qwen3-VL-8B-Instruct	29.5	28.2	32.1	33.8	28.0	31.0	27.4	25.7
Qwen3-VL-4B-Instruct	3.3	3.7	3.5	3.3	3.0	4.0	2.7	3.2
Molmo2-8B	15.5	17.5	15.3	14.8	15.5	17.5	16.7	11.5
Molmo2-4B	10.9	14.3	10.9	13.5	11.7	11.8	8.5	5.3
Gemma-3-12B-IT	18.7	22.9	16.4	20.1	17.2	16.1	17.9	20.4
Gemma-3-4B-IT	7.2	8.3	6.9	6.4	6.8	7.6	7.5	6.7
SEA-LION-v4-8B-VL	33.3	33.4	33.8	32.1	33.9	36.4	34.5	29.1
SEA-LION-v4-4B-VL	10.0	11.6	9.2	9.9	9.6	11.4	9.8	8.9
qwen-qwen3-5-27b	51.2	57.0	49.2	48.2	49.1	53.9	47.8	53.0

E ROBUSTNESS OF THE STEP-LEVEL JUDGE

Using GPT-5.4 to judge its own reasoning steps raises a concern about self-preference bias, where an LLM judge favors responses generated by itself. This concern applies only to the step-level evaluation in Table 2. All accuracy results (Tables 1, 11, and 12) are computed by matching the final answer against the ground truth with a 1% tolerance, without any model involved. The judging task is also narrower than the open-ended comparisons where self-preference is usually observed. Since every template comes with a gold solution, the judge only decides whether a gold step is covered by the model output, or whether a model step is valid against the gold solution (Tables 9 and 10).

To verify that our step-level findings hold regardless of the judge, we re-evaluated the same GPT-5.4 outputs using Gemini 2.5 Pro as the judge with identical prompt templates. Table 13 compares the results. Scores under Gemini 2.5 Pro are slightly lower in most subjects, with the largest difference in Computer Science, where $\mathcal{F}$ drops by 4.6 points and $\mathcal{F}_{wst}$ by 9.1 points. Nevertheless, the gap between $\mathcal{F}$ and $\mathcal{F}_{wst}$ remains wide in every subject under both judges (13.0–20.8 points with GPT-5.4 and 15.5–25.0 points with Gemini 2.5 Pro), and Biology remains the most challenging subject. This indicates that the reasoning step fragility reported in Section 4.2 holds under a judge from a different model family.

Table 13: Step-level evaluation of GPT-5.4 outputs judged by GPT-5.4 and Gemini 2.5 Pro. Δ denotes the change when switching the judge from GPT-5.4 to Gemini 2.5 Pro.

Subject	$\mathcal{F}$			$\mathcal{F}_{wst}$		
	GPT-5.4	Gemini-2.5-Pro	Δ	GPT-5.4	Gemini-2.5-Pro	Δ
Mathematics	88.0	86.5	-1.5	67.8	66.5	-1.3
Physics	92.2	92.4	+0.2	79.2	76.7	-2.5
Chemistry	93.4	93.2	-0.2	79.0	77.7	-1.3
Biology	81.1	80.0	-1.1	60.3	55.0	-5.3
Computer Science	89.4	84.8	-4.6	75.1	66.0	-9.1