Can Vision-Language Models Evaluate Handwritten Math?

Oikantik Nath[1,2] **Hanani Bathina**[2,3]
Mohammed Safi Ur Rahman Khan[1,2] **Mitesh M. Khapra**[1,2]
[1]Indian Institute of Technology, Madras [2]AI4Bharat [3]Chennai Mathematical Institute
Correspondence: {oikantik, miteshk}@cse.iitm.ac.in, hananib.mds2024@cmi.ac.in
🤗 https://huggingface.co/datasets/ai4bharat/FERMAT
https://github.com/AI4Bharat/FERMAT

Abstract

Recent advancements in Vision-Language Models (VLMs) have opened new possibilities in automatic grading of handwritten student responses, particularly in mathematics. However, a comprehensive study to test the ability of VLMs to evaluate and reason over handwritten content remains absent. To address this gap, we introduce **FERMAT**, a benchmark designed to assess VLMs' ability to detect, localize and correct errors in handwritten mathematical content. **FERMAT** spans four key error dimensions - computational, conceptual, notational, and presentation - and comprises over 2,200 handwritten math solutions derived from 609 manually curated problems from grades 7-12 with intentionally introduced perturbations. Using **FERMAT** we benchmark nine VLMs across three tasks: error detection, localization, and correction. Our results reveal significant shortcomings in current VLMs in reasoning over handwritten text, with GEMINI-1.5-PRO achieving the highest error correction rate (77%). We also observed that some models struggle with processing handwritten content, as their accuracy improves when handwritten inputs are replaced with printed text or images. *These findings highlight the limitations of current VLMs and reveal new avenues for improvement.* We release **FERMAT** and all the associated resources in the open-source to drive further research.

1 Introduction

Recent advancements in Large Language Models (LLMs) (Jiang et al., 2023; Touvron et al., 2023; Yang et al., 2024; Anil et al., 2023) and Vision-Language Models (VLMs) (Team et al., 2024; Dubey et al., 2024; OpenAI, 2024; Wang et al., 2024b; Agrawal et al., 2024; Liu et al., 2024) have significantly enhanced the ability to interpret both textual and visual data. These developments are driving progress in core language (Zhao et al., 2023; Xinyi et al., 2023; Wang et al., 2024d) and vision-language tasks (Zhang et al., 2024; Li et al., 2023a), with notable advancements in mathematical reasoning and problem-solving (Frieder et al., 2023; Lu et al., 2023b; Shao et al., 2024; Zhou et al., 2024; Imani et al., 2023; Bang et al., 2023). As these models evolve, they are increasingly enabling sophisticated applications in educational tools (Wang et al., 2024e), including automated evaluation (Malik et al., 2021; Li et al., 2024a; Sonkar and Baraniuk, 2023; Tigina et al., 2023), quiz generation (Li et al., 2024a; Scaria et al., 2024), and personalized tutoring systems (Wang et al., 2024c; Alhafni et al., 2024; Abu-Rasheed et al., 2024; Li et al., 2023b).

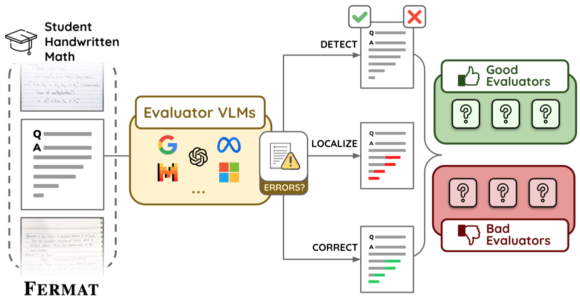

Figure 1: We introduce **FERMAT**, a novel multimodal benchmark to evaluate VLMs on their ability to detect, reason about, and assess the correctness of handwritten grade-school level math solutions.

One promising application of VLMs is exemplified by OpenAI's widely referenced demo[1], which demonstrated the potential of such models to evaluate handwritten math content produced by students. This requires a model to accurately understand, identify, and correct potential errors. Although these demonstrations highlight potential, a robust and comprehensive evaluation of VLMs for this task remains lacking. To address this gap, a benchmark analogous to Checklist-based fine-grained assessments for text (Ribeiro et al., 2020; Zhou et al., 2024; Sonkar and Baraniuk, 2023) is essential.

[1]https://www.youtube.com/watch?v=_nSmkyDNulk

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14784–14814
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

To address this need, we introduce **FERMAT**, a benchmark to evaluate a VLM's capability in **F**inding and correcting **ER**rors in handwritten **MAT**hematical content. This benchmark enables the evaluation of Vision-Language Models (VLMs) as automatic evaluators for handwritten math responses across four common error axes: (a) computational errors, (b) conceptual misunderstandings, (c) notation errors, and (d) presentation issues. To accomplish this, we first manually curated 609 math problems from grades 7 to 12, along with their correct solutions. We then used a human-in-the-loop approach to introduce targeted perturbations into these correct solutions along the previously defined error axes. Finally, these perturbed solutions were transcribed by more than 40 human annotators to produce handwritten versions. The transcriptions reflect natural variations in handwriting styles, and the captured images reflect differences in lighting, paper types, and overall image quality. The resulting benchmark contains more than 2200 handwritten *erroneous* math solutions and their corresponding correct "gold" answers in LaTeX format.

Using **FERMAT**, we evaluate nine VLMs on three core tasks: (a) Error Detection, (b) Error Localization, and (c) Error Correction. Our experiments show that most models struggle with these tasks, with GEMINI-1.5-PRO leading with the best performance of 77% in Error Correction. We also find that providing additional meta-information about the problem type, grade level, error category, etc. improves model performance. Furthermore, our analysis shows that Error Localization accuracy increases when handwritten inputs are replaced with printed images or direct text, highlighting the challenges in processing handwritten content. Overall, these findings highlight key limitations in modern VLMs when processing handwritten mathematical content, emphasizing the need for caution in real-world applications.

2 Related Work

Multimodal Evaluations. The evaluation of VLMs across different multimodal tasks has garnered significant attention in recent works. Prior works (Zhang et al., 2023; Yue et al., 2024a; Das et al., 2024; Yue et al., 2024b; Zhong et al., 2023) have introduced multi-disciplinary benchmarks using questions from different competitive exams. Additionally, reasoning benchmarks, including mathematical (Mishra et al., 2022; Lu et al., 2023a; Wang et al., 2024a) and broader STEM-oriented benchmarks (He et al., 2024), have been widely explored. While most existing studies evaluate images paired with simple typed text, Liu et al. (2023) and Bubeck et al. (2023) investigate OCR capabilities for handwritten text, focusing on single-line mathematical expressions. In contrast, our benchmark includes dense, handwritten, multi-line derivations and complex mathematical notations, hence providing a more rigorous evaluation.

Error Evaluation Abilities of LLMs. Prior studies (Kamoi et al., 2024; Doddapaneni et al., 2024; An et al., 2023) have explored LLMs' ability to detect textual errors. Some works (Li et al., 2024b; Tyen et al., 2024; Sonkar and Baraniuk, 2023) highlight that, although LLMs struggle with error detection in mathematical text, they show strong correction abilities. While most research has focused on text-based contexts, a few works (Yan et al., 2024; Zhou et al., 2024) examine multimodal error detection, primarily targeting simple objective errors. In contrast, our benchmark introduces a more realistic evaluation, including multiple variations of a single error type, resulting in a deeper assessment of VLMs' ability to identify and correct complex multimodal mathematical errors.

CHECKLIST-inspired Work. The CHECKLIST framework (Ribeiro et al., 2020) established a systematic approach for evaluating NLP models via behavioral testing. Its principles have been adapted for LLM evaluations, such as FBI (Doddapaneni et al., 2024), MATHCHECK (Zhou et al., 2024), and DUPE (Sonkar and Baraniuk, 2023), with a focus on robustness by introducing controlled perturbations in the outputs. Building on this foundation, we introduce a tailored perturbation taxonomy for evaluating handwritten error detection, localization and correction ability of different VLMs.

3 FERMAT Benchmark

We present **FERMAT**, a benchmark of 2,244 handwritten solved math problems spanning middle and high school topics, including Arithmetic, Algebra, Mensuration, Geometry, Probability, Statistics, Trigonometry and Calculus. Each solution reflects common mistakes made by students across four different axes: (i) computational errors, (ii) conceptual misunderstandings, (iii) notation errors, and (iv) presentation issues. Additionally, we also include some superficial perturbations that do not render the solution incorrect (e.g., "16 cm" vs. "16.0

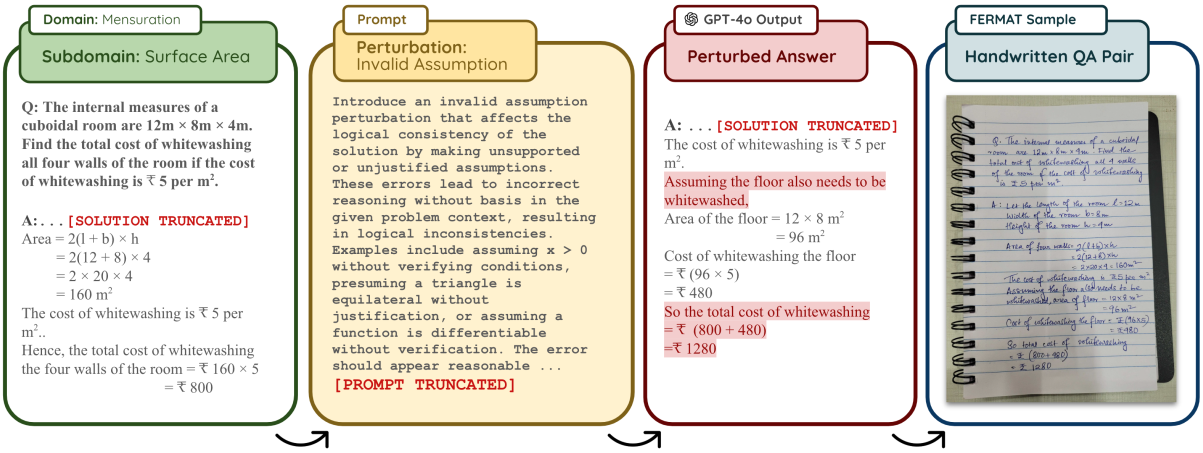

Figure 2: The construction of **FERMAT** involves four steps: (1) sampling problems with detailed solutions from math domains (§3.1), (2) defining a perturbation taxonomy (§3.2), (3) applying perturbations to solutions (§3.3), and (4) transcribing the perturbed QA pairs (§3.4).

cm”). Each instance in **FERMAT** comprises a tuple (Q, I_{hw}, A_{gold}), where Q represents the question, I_{hw} denotes the image containing the handwritten question and the *erroneous* solution, and A_{gold} is the original correct solution of Q. Both Q and A_{gold} are provided in LaTeX to ensure standard uniform representation across different benchmarks.

The introduced errors are based on well-defined axes of commonly occurring errors designed to rigorously test multimodal reasoning and auto-evaluation capabilities of VLMs. A detailed description of these axes can be found in Table 2. To ensure high standards and sanctity of the benchmark, each instance undergoes multiple stages of manual vetting, from problem-set curation (§3.1), defining different error categories (§3.2), creating perturbations (§3.3), to manually transcribing and verifying the perturbed handwritten answers (§3.4).

3.1 Problem Set Collection

Initial Data Collection We first manually collect well-formulated solved problems from widely recognized math textbooks commonly used in grades 7 to 12 curricula. These problems and their solutions are extracted as images from these textbooks, ensuring a diverse representation of core mathematical domains, including Arithmetic, Algebra, Mensuration, Geometry, Probability, Statistics, Trigonometry, and Calculus. This approach ensures comprehensive coverage of foundational concepts across middle and high school levels.

This initial problem set includes only problems with detailed free-form solutions. To enhance the diversity of question formats, we also include multiple-choice questions (MCQs) along with their solutions. These MCQs, sourced from various competitive exams, cover key topics in Quantitative Aptitude, such as profit and loss, time and work, and data interpretation. These topics often involve practical applications of mathematical concepts often underrepresented in standard textbooks.

LaTeX Conversion and Verification After collecting around 850 diverse problem-solution images, we used GPT-4o to extract the content in LaTeX format. We choose GPT-4o over standard OCR engines due to its superior capability in handling complex mathematical notations (Kaltchenko, 2024) and its ability to give well-formatted outputs. All the extracted LaTeX content was then rigorously reviewed by the authors for correctness, resulting in 609 high quality LaTeX problem-solution pairs (Q, A_{gold}), spanning more than 50 fine-grained topics across the above mentioned 7 domains.

3.2 Designing the Perturbation Taxonomy

To reflect common mistakes made by students, we manually designed a comprehensive taxonomy of perturbations specific to our mathematical domains. These perturbations, introduced into correct solutions, are categorized into five broad axes:

Computational Errors (CO): Errors made in different computations, such as arithmetic mistakes in intermediate or final steps.

Conceptual Errors (CP): Errors made while incorrectly applying concepts, including misinterpretations (e.g., solving for area instead of perimeter) or misuse of identities, like $(a + b)^2 = a^2 + b^2$.

Notational Errors(NO): Errors made by incor-

Category	# Instances
TOTAL NUMBER OF QUESTIONS	**2,244**
Free-Form Question-Answer Pairs	1,814 (82%)
MCQs with Free-Form Explanations	430 (18%)
DOMAINS (# SUBDOMAINS)	
Algebra (11)	686 (28.6%)
Aptitude (1)	430 (17.9%)
Arithmetic (13)	500 (20.9%)
Calculus (8)	305 (12.8%)
Mensuration and Geometry (11)	260 (10.9%)
Probability and Statistics (4)	109 (4.6%)
Trigonometry (2)	101 (4.2%)
GRADE LEVELS	**7 - 12**
Total Number of Annotators	43
Average Annotations per Annotator	55.6

Table 1: Key statistics of **FERMAT**. Subdomains and perturbation versus grade are detailed in Appendix A.

rect usage of symbols, operators, or formulae, such as writing x^2 as $x2$ or substituting $+$ for $\times$.

Presentation Errors(PR): Clarity or formatting issues, such as providing an answer in fraction form when a decimal is requested, or using inconsistent terminology (e.g., switching between "vector" and "line") that may cause contextual confusion.

Superficial Perturbations (SU): Non-impactful errors made by making subtle changes, such as superficially altering variable names ($f(x) = x^2$ to $f(t) = t^2$) or omitting non-essential intermediate steps without affecting solution correctness. These errors evaluate the VLMs' ability to maintain evaluation accuracy despite superficial modifications.

A detailed description of each error axis and perturbations are provided in Table 2. For each of these, the VLM is expected to detect, and correct errors accurately, while ignoring the superficial perturbations that do not affect the solution's validity.

3.3 Human-In-The-Loop Perturbation Generation

Based on the perturbation taxonomy (§3.2), a subset of relevant perturbations is manually selected for each math domain. For each problem curated in (§3.1), a domain-specific perturbation is applied using GPT-4o, denoted as $f(\cdot)$, by prompting it with the LaTeX question Q and correct solution A_{gold}. This process is represented as $f(P, X_P, Q, A_{gold}) \rightarrow (exp, A_{pert})$, where P is the chosen perturbation, X_P represents instructions for inducing the perturbation along with three in-context examples, A_{pert} is the perturbed solution, and exp explains the introduced perturbation. This process is repeated until all problems undergo the relevant perturbations within its domain's subset, ensuring comprehensive coverage.

While GPT-4o generally produces the intended perturbations, occasional inconsistencies are observed, such as deviations from the specified perturbation, irrelevant modifications, misaligned reasoning, or unchanged answers despite correct reasoning. To address these issues, all perturbed answers (A_{pert}) are manually verified by the authors to ensure that intended perturbation is correctly applied and that the reasoning aligns with it. During this review process, the induced perturbations are further classified as true errors or superficial changes. Further details of this are provided in Appendix A.2.

3.4 Handwritten Transcription with Manual Verification

We engaged a team of 43 annotators from diverse demographic backgrounds to manually transcribe each perturbed answer A_{pert}. Annotators were instructed to use various paper types and colored pens or inks. The handwritten questions and solutions were captured using mobile phone cameras by the annotators and subsequently uploaded to a centralized portal. This process ensured a diverse benchmark, reflecting a wide range of handwriting styles, paper types, and lighting conditions. As each problem underwent multiple perturbations, the dataset effectively simulates exam-like scenarios where students encounter similar questions but make distinct mistakes in their responses.

Each image I_{hw} was then manually verified by the authors to ensure correct replication of the intended perturbation. During this verification, we recorded additional metadata such as handwriting legibility, image orientation, and overall image quality for each I_{hw}. A custom validation tool was developed to streamline this review and annotation process. Detailed statistics on **FERMAT** are provided in Table 1, and further details on the verification tool in Appendix B.

4 Evaluation Setup

In this section, we outline the different tasks on which we evaluate different VLMs on **FERMAT**. Each VLM, denoted by $f(\cdot)$, takes as input a handwritten answer I_{hw} (§3.4) and a prompt P_x specific to a task x. Detailed prompts for all tasks are pro-

Perturbation Axes	# Inst	Perturbation Description
COMPUTATIONAL (CO)	611	CALCULATION & PROPAGATION ERRORS
FINAL NUMBER	156	Incorrect final answer including digit swaps or misplaced decimals.
INTERMEDIATE CALCULATION	100	Arithmetic calculation errors in intermediate steps.
NON-PROPAGATED STEP ERROR	80	Error in intermediate step corrected in subsequent steps.
PROPAGATED STEP ERROR	108	Error in intermediate step carried forward.
COPY ERROR	167	Copying wrong numbers/expressions from question (e.g., copying 45 as 54).
CONCEPTUAL (CP)	**609**	INCORRECT INTERPRETATION OF CONCEPTS
THEOREM MISAPPLICATION	62	Applying theorems/identities incorrectly (e.g., using $\sin^2\theta + \cos^2\theta = 0$).
MISINTERPRET QUESTION	145	Misreading problem requirements such as reporting area instead of volume.
INVALID ASSUMPTION	122	Making assumptions without justification/verification.
OUTRIGHT INCORRECT FACT	143	Stating objectively false information (e.g., a triangle has two right angles).
FORMULA MISUSE	137	Incorrectly writing a standard formula (e.g., Circle Area: $\pi r^2 \rightarrow 2\pi r$).
NOTATIONAL (NO)	**255**	MISTAKES IN MATH SYMBOLS & OPERATORS
SYMBOL ERROR	81	Mistakes in symbols/notation (e.g., $x^2 \rightarrow x2$).
OPERATOR SWAP	115	Incorrect substitution of operators (e.g., $+ \rightarrow \times$).
MISPLACED PARENTHESES	59	Misplacing parentheses, thus changing the intended order of operations.
PRESENTATION (PR)	**429**	ISSUES IN FORMATTING & LOGICAL FLOW
FORMAT IGNORED	47	Ignoring question-specified format (e.g., standard vs scientific notation).
TERMINOLOGY SWAP	25	Switching inconsistently between terms (e.g., "vector" $\longleftrightarrow$ "line").
LOGIC DISRUPTION	101	Presenting steps out of logical order (e.g., final answer used in earlier steps).
CONTEXTUAL SWAP	43	Contextually similar but incorrect term substitution (e.g., circle $\rightarrow$ ellipse).
VARIABLE MISNAMING	67	Swapped variables (e.g., swapping a and b in a quadratic formula).
INCORRECT UNITS	146	Reporting with wrong units (e.g., length in kg instead of m).
SUPERFICIAL (SU)	**340**	MODIFICATIONS WITHOUT IMPACTING CORRECTNESS
SUPERFICIAL VAR CHANGE	100	Superficially changing variable names (e.g., $f(x) = x^2 \rightarrow f(t) = t^2$).
STEP OMISSION	81	Skipping non-essential intermediate steps.
IRRELEVANT INFO	159	Including unnecessary information (e.g., adding unrelated discussions).
TOTAL INSTANCES	**2244**	

Table 2: Overview of perturbation categories with descriptions for perturbation. Correct original text is highlighted in green, while perturbed text is highlighted in red.

vided in Appendix F. We propose three tasks of increasing difficulty: (i) Error Detection (§4.1), (ii) Error Localization (§4.2), and (iii) Error Correction (§4.3). For each task, we evaluate multiple strategies, all using a Chain-of-Thought (COT) (Wei et al., 2022) method, by asking the VLM to provide a step-by-step reasoning before giving its answer.

4.1 Error Detection

In this task, the VLM $f(\cdot)$ is prompted to detect the error in the given handwritten image I_{hw} and give a binary output indicating the presence of an error along with its reasoning.

ED: In this strategy, the VLM, $f(\cdot)$ is directly provided with a handwritten image I_{hw} and a prompt (P_{ED}) to detect the error and output a binary value ($True/False$), indicating the presence of an error in the solution and a reasoning (exp) for the same. We denote this formally as $f(P_{ED}, I_{hw}) \rightarrow (exp, True/False)$.

ED+OCR: In this strategy, we decompose the task into two steps, where first the VLM, $f(\cdot)$, is provided with the handwritten image I_{hw} and prompt (P_{OCR}) to perform OCR and convert the handwritten content into LaTeX format. Next, the same VLM, $f(\cdot)$, is prompted with the resulting LaTeX text, to detect the error and output a binary value ($True/False$) along with the reason. This is formally denoted as $f(P_{ED}, f(P_{OCR}, I_{hw})) \rightarrow (exp, True/False)$.

4.2 Error Localization

In this task, the VLM, $f(\cdot)$, is prompted to accurately localize the error in the given handwritten image I_{hw}, by identifying the specific line where the error occurs and providing reasoning for its decision. If no error is present, then the model is asked to output "NA" (Not Applicable). This task is more challenging than error detection (ED) (§4.1) since the VLM must perform both error detection and localization simultaneously.

EL: In this strategy, the VLM, $f(\cdot)$ is directly given a handwritten image I_{hw} along with a prompt (P_{EL}) to localize the error, if present, in the image. The VLM describes the specific line(s) containing the error(s) and provides an explanation. Formally, this is represented as $f(P_{EL}, I_{hw}) \rightarrow (exp, text_{loc}/NA)$.

EL+OCR: Similar to the **ED+OCR** strategy discussed in Sec §4.1, the VLM $f(\cdot)$ is first prompted to perform OCR on the given handwritten image I_{hw} and then asked to localize the error in the output LaTeX text by describing the specific line(s) containing the error(s). This is formally denoted as $f(P_{EL}, f(P_{OCR}, I_{hw})) \rightarrow (exp, text_{loc}/NA)$.

4.3 Error Correction

In this task, the VLM $f(\cdot)$ is prompted to correct any errors found in a given handwritten image I_{hw} and output the entire corrected solution in LaTeX format. If no error is present, the VLM is asked to output "NA". This is the most challenging of the three tasks, as the VLM must perform error detection, localization, and correction in a single step.

EC: In this strategy, the VLM, $f(\cdot)$, is directly given the handwritten image I_{hw} along with the prompt (P_{EC}) to correct any errors. If errors are detected, the model outputs the entire corrected solution A_{corr}, otherwise, it returns "NA" to indicate the solution is already correct. Since a problem can often be solved in multiple different ways to reach the final answer, the model is allowed to explore all possible ways to generate the correct answer to the problem. The error correction strategy is formally denoted as $f(P_{EC}, I_{hw}) \rightarrow (exp, A_{corr}/NA)$.

EC+OCR: Similar to the strategies discussed in Sec §4.1 and §4.2, the VLM $f(\cdot)$ is first prompted to perform OCR on the given image and then prompted to give the entire corrected answer or "NA" if no error is found. Formally, we represent this process as $f(P_{EC}, f(P_{OCR}, I_{hw})) \rightarrow (exp, A_{corr}/NA)$.

4.4 Cascaded Setup

In the above setups, each of the three tasks was performed independently. Here, we evaluate a cascaded setup where these tasks are executed sequentially, as shown in Figure 3. In this approach, the VLM $f(\cdot)$ first performs error detection as outlined in **ED** (§4.1). For images identified as containing errors, error localization is then performed using **EL** (§4.2). Finally, for images with localized errors, the error correction step is executed based on the method described in **EC** (§4.3). Unlike previous setups, the output of each stage is passed as input to the next. For example, during error correction, the VLM is provided with both the original image and the localized error line(s) from the previous step to improve accuracy. The cascaded setup aims to achieve precise error correction by leveraging the context generated at each stage. Formally, this process can be represented as $f(P_{EC}, I_{hw}, f(P_{EL}, I_{hw}, f(P_{ED}, I_{hw}))) \rightarrow (exp, A_{corr}/NA)$.

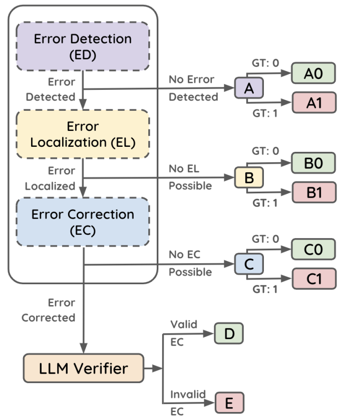

Figure 3: Cascaded black-box evaluation setup, as described in §4.4.GT denotes Ground Truth. The total number of correctly evaluated **FERMAT** samples in this setup is represented by the summation of $A0$, $B0$, $C0$, and D.

4.5 LLM as an Evaluator

Error localization (§4.2) and correction (§4.3) are inherently subjective tasks, as multiple valid solutions can exist. While human evaluation remains the gold standard for VLM assessment, it is costly and time-intensive. To address this, we use Large Language Models (LLMs) as automated evaluators, following recent advancements (Zheng et al., 2023; Chiang and Lee, 2023). For localization, the LLM checks if errors are correctly identified, and for correction, it verifies the accuracy of the corrected solution. We use GPT-4O as our Evaluator LLM due to its widespread use as an evaluator.

To validate the reliability of our GPT-4O-based Evaluator LLM, we conducted a study on 464 randomly sampled task outputs from four VLMs: GPT-4O, LLAMA-3.2-11B, PIXTRAL-12B, and PHI-3.5-VI. Graduate students were independently

Models	Non-cascaded						Cascaded
	ED	ED+OCR	EL	EL+OCR	EC	EC+OCR	ED ▸ EL ▸ EC
	BACC	BACC	ACC	ACC	ACC	ACC	ACC
GEMINI-1.5-PRO	0.63	**0.67**	0.43	**0.56**	**0.76**	**0.77**	0.50
GEMINI-1.5-FLASH	0.60	0.62	0.39	0.51	0.70	0.72	0.46
GPT-4O	**0.65**	0.64	**0.45**	0.50	0.66	0.71	0.45
GPT-4O-MINI	0.55	0.57	0.44	0.45	0.56	0.58	**0.51**
LLAMA-3.2-90B	0.52	0.62	0.18	0.41	0.25	0.57	0.31
LLAMA-3.2-11B	0.50	0.52	0.14	0.27	0.21	0.38	0.20
PIXTRAL-124B	0.52	0.59	0.24	0.40	0.46	0.56	0.26
PIXTRAL-12B	0.51	0.55	0.24	0.27	0.30	0.34	0.32
PHI-3.5-VI	0.52	0.51	0.06	0.09	0.15	0.12	0.11

Table 3: Performance comparison of VLMs in cascaded and non-cascaded settings on **FERMAT** across different evaluation strategies. Metrics include Balanced Accuracy (**BACC**) for error detection, and Accuracy (**ACC**) for error localization and correction. Higher values (↑) indicate better performance.

tasked with assessing the VLM outputs to determine their accuracy. We then compared these human judgments with the evaluations produced by our Evaluator LLM and found a *94%* average agreement between the two. Given this strong alignment with human evaluations, we opted to use our GPT-4O based Evaluator LLM as a faster but equally reliable alternative to the expensive and time-consuming human evaluations for all subsequent experiments. The prompts used for our Evaluator LLM as well as details about the human verification are provided in Appendix C and F.

5 Experiments

We evaluate nine popular VLMs, including both closed-proprietary and open-sourced models as listed in Table 3 on **FERMAT**. For each task, we ensure consistent evaluation by using identical prompts across all models and setting the sampling temperature to zero to maintain reproducibility. Similarly, for the Evaluator LLM, we use GPT-4O with a temperature of zero. Detailed prompts for all the experiments are provided in Appendix F.

For the Error Detection task (§4.1), we report the model performance using Balanced Accuracy, which accounts for the class imbalance by averaging the *sensitivity* (true positive rate) and *specificity* (true negative rate). Ground truth labels are defined as 0 for Superficial Perturbations (SU) (§3.2) and 1 for all other error types. We report Balanced Accuracy instead of the standard F1 score since it gives equal importance to both positive and negative labels, whereas the F1 score ignores the true negatives altogether. We provide additional information regarding F1 and Accuracy scores in Appendix D.

For the Error Localization (§4.2) and Error Correction (§4.3) tasks, we report Accuracy, which we define as the proportion of times the Evaluator LLM (§4.5) determines that the VLM has done an accurate job.

5.1 How do different VLMs perform?

We present the main results of our tasks in Table 3. Overall, all models face challenges in the core tasks of **FERMAT**, with GPT-4O and GEMINI-1.5-PRO consistently leading across all tasks. GPT-4O demonstrates superior performance in the **ED** and **EL** tasks, while GEMINI-1.5-PRO achieves the best results in the remaining tasks. Most models perform well on the Error Detection task, but performance declines significantly as task complexity increases for Localization and Correction. A detailed analysis of this trend is provided in Table 7. We also observe that introducing an explicit OCR step, improves performance for certain models. Notably, PIXTRAL-124B and LLAMA-3.2-90B show large gains, which can be attributed to stronger handwriting OCR capabilities compensating for weaker multimodal reasoning. By contrast, models with strong multimodal understanding, such as GPT-4O and GEMINI-1.5-PRO, gain marginal benefits from the OCR step, suggesting they rely less on textual signals and are better at jointly interpreting visual and textual content.

5.2 How do VLMs perform in the Cascaded Approach?

We evaluate all models in the cascaded setup described in §4.4. As shown in the last column of Table 3, decomposing the Error Correction task into

Model	Base	L1	L2	L3	L4
GPT-4o	0.658	0.670	0.676	0.691	0.702

Table 4: **BACC** (Balanced Accuracy) scores of GPT-4o on the error detection task under increasing levels of helpful contextual information included in the prompt. Higher scores indicate better performance.

sequential steps leads to a significant performance drop across models, including GPT-4o, Gemini-1.5-Flash, Gemini-1.5-Pro, and Pixtral-124B. This decline is primarily attributed to the cautious error detection behavior of these models (discussed in Table 7), which results in a large proportion of images being filtered out during the initial stage (**ED**). A comprehensive breakdown of intermediate and final outputs for each VLM in the cascaded setup is provided in Appendix E.

5.3 Does more information help VLMs?

We conducted an ablation study to evaluate whether providing additional information about the error type improves model performance. Four settings with increasing levels of information were designed: **L1** (basic context, including grade, math domain, and subdomain), **L2** (L1 + descriptions of *all* perturbations specific to that domain + some examples of perturbations), **L3** (L1 + specifying the *exact* perturbation category that was applied), and **L4** (L3 + a sample *erroneous* solution accompanied by an explanation of the mistake). As shown in Table 4, performance consistently improves with the addition of more detailed information, indicating that increasing error context facilitates better Error Detection. Prompts designed for this study are provided in Appendix F.

We note that while this experiment provides valuable insights from an ablation perspective, incorporating such detailed information may be challenging in practical scenarios. For example, if a teacher is required to specify the exact error type in a solution, they might find it more practical to evaluate the solution directly without relying on a VLM.

5.4 How much does handwriting affect model performance on Fermat ?

We hypothesize that weaker handwriting recognition capabilities in some models (Table 3) may impair their ability to identify and correct mistakes. To test this, we conduct two studies to isolate reasoning abilities from visual processing. First, we replace handwritten images with printed LaTeX rendered images from the (Q, $A_{perturb}$) pairs (§3.3). Second, we eliminate images entirely, providing direct LaTeX text inputs for Q and $A_{perturb}$. As shown in Figure 4, performance improves consistently as visual complexity is reduced. The largest gains occur when switching to text input while replacing handwritten images with printed LaTeX still offers small benefits on an average. These results highlight the challenges of processing handwritten content and reinforce **Fermat**'s rigor as a benchmark for evaluating both reasoning and visual understanding in VLMs.

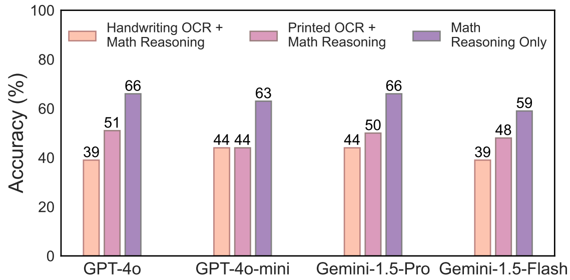

Figure 4: Performance of VLMs on the error localization task across various benchmark settings. Higher scores (↑) indicate better performance.

6 Conclusion

We introduce **Fermat**, a comprehensive benchmark to assess Vision-Language Models (VLMs) on their ability to detect, localize, and correct errors in handwritten mathematical content. By spanning four critical error dimensions — computational, conceptual, notational, and presentation — and curating over 2,200 perturbed handwritten solutions from 609 math problems (grades 7-12), **Fermat** provides a robust evaluation framework. Our analysis of nine prominent VLMs reveals key limitations in their reasoning over handwritten content. While Gemini-1.5-Pro achieves the highest error correction rate (77%), we find that smaller models often struggle. Our findings also highlight the challenges posed by handwritten content, as models perform better with printed images or text inputs. By releasing **Fermat** and all associated resources as open-source, we hope that this fosters further research on evaluating and enhancing the capabilities of VLMs for real-world applications.

Limitations

While we have compiled a comprehensive list of perturbation categories, we acknowledge that it may not be exhaustive, leaving room for further expansion. Our benchmark primarily focuses on school-level mathematics questions, with more advanced topics and question types left for future work. Additionally, we do not explore complex multi-agent approaches for error detection, instead limiting our study to single or dual LLM calls.

Ethics Statement

Annotators who participated in the annotation and/or verification task are paid a competitive monthly salary to help with the tasks. The salaries were determined based on the qualification and the prior experience working on similar tasks and adhering to the norms of the government of our country. The annotators were made aware that the datasets will be publicly released. The annotated datasets have no personally identifying information. The datasets used in this paper will be made available under permissible licenses, and we adhere strictly to their intended usage, maintaining compliance with licensing requirements. Additionally, all the code used for our evaluations and perturbation generation will be made publicly available under the MIT License. We only used AI Assistants for assistance purely with the language of the paper, e.g., paraphrasing, spell-checking, or polishing the author's original content, without suggesting new content

References

Hasan Abu-Rasheed, Christian Weber, and Madjid Fathi. 2024. Knowledge graphs as context sources for llm-based explanations of learning recommendations. *arXiv preprint arXiv: 2403.03008*.

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, Jessica Chudnovsky, Diogo Costa, Baudouin De Monicault, Saurabh Garg, Theophile Gervet, Soham Ghosh, Amélie Héliou, Paul Jacob, Albert Q. Jiang, Kartik Khandelwal, Timothée Lacroix, Guillaume Lample, Diego Las Casas, Thibaut Lavril, Teven Le Scao, Andy Lo, William Marshall, Louis Martin, Arthur Mensch, Pavankumar Muddireddy, Valera Nemychnikova, Marie Pellat, Patrick Von Platen, Nikhil Raghuraman, Baptiste Rozière, Alexandre Sablayrolles, Lucile Saulnier, Romain Sauvestre, Wendy Shang, Roman Soletskyi, Lawrence Stewart, Pierre Stock, Joachim Studnia, Sandeep Subramanian, Sagar Vaze, Thomas Wang, and Sophia Yang. 2024. Pixtral 12b. *arXiv preprint arXiv: 2410.07073*.

Bashar Alhafni, Sowmya Vajjala, Stefano Bannò, Kaushal Kumar Maurya, and Ekaterina Kochmar. 2024. Llms in education: Novel perspectives, challenges, and opportunities. *arXiv preprint arXiv: 2409.11917*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv: 2310.20689*.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *arXiv preprint arXiv: 2305.10403*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *International Joint Conference on Natural Language Processing*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter

Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv: 2303.12712*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.

Rocktim Jyoti Das, Simeon Emilov Hristov, Haonan Li, Dimitar Dimitrov, Ivan Koychev, and Preslav Nakov. 2024. EXAMS-V: A multi-discipline multilingual multimodal exam benchmark for evaluating vision language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7768–7791. Association for Computational Linguistics.

Sumanth Doddapaneni, Mohammed Safi Ur Rahman Khan, Sshubam Verma, and Mitesh M. Khapra. 2024. Finding blind spots in evaluator llms with interpretable checklists. *Conference on Empirical Methods in Natural Language Processing*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, Alexis Chevalier, and J. Berner. 2023. Mathematical capabilities of chatgpt. *Neural Information Processing Systems*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv: 2402.14008*.

Shima Imani, Liang Du, and H. Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *Annual Meeting of the Association for Computational Linguistics*.

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

Alexei Kaltchenko. 2024. Assessing gpt model uncertainty in mathematical ocr tasks via entropy analysis.

Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Ranran Haoran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. 2024. Evaluating llms at detecting errors in llm responses. *arXiv preprint arXiv: 2404.03602*.

Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. 2023a. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends in Computer Graphics and Vision*.

Hang Li, Tianlong Xu, Chaoli Zhang, Eason Chen, Jing Liang, Xing Fan, Haoyang Li, Jiliang Tang, and Qingsong Wen. 2024a. Bringing generative ai to adaptive learning in education. *arXiv preprint arXiv: 2402.14601*.

Qingyao Li, Lingyue Fu, Weiming Zhang, Xianyu Chen, Jingwei Yu, Wei Xia, Weinan Zhang, Ruiming Tang, and Yong Yu. 2023b. Adapting large language models for education: Foundational capabilities, potentials, and challenges. *arXiv preprint arXiv: 2401.08664*.

Xiaoyuan Li, Wenjie Wang, Moxin Li, Junrong Guo, Yang Zhang, and Fuli Feng. 2024b. Evaluating mathematical reasoning of large language models: A focus on error identification and correction. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11316–11360, Bangkok, Thailand. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2023. Ocrbench: On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv: 2305.07895*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023a. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv: 2310.02255*.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023b. A survey of deep learning for mathematical reasoning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14605–14631, Toronto, Canada. Association for Computational Linguistics.

Ali Malik, Mike Wu, Vrinda Vasavada, Jinpeng Song, Madison Coots, John C. Mitchell, Noah D. Goodman, and C. Piech. 2021. Generative grading: Near human-level accuracy for automated feedback on richly structured problems. *Educational Data Mining*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5807–5832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2024. `https://openai.com/index/hello-gpt-4o/`.

Marco Tulio Ribeiro, Tongshuang Sherry Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *Annual Meeting of the Association for Computational Linguistics*.

Nicy Scaria, S. Chenna, and Deepak N. Subramani. 2024. Automated educational question generation at different bloom's skill levels using large language models: Strategies and evaluation. *International Conference on Artificial Intelligence in Education*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv: 2402.03300*.

Shashank Sonkar and Richard G. Baraniuk. 2023. Deduction under perturbed evidence: Probing student simulation capabilities of large language models. *arXiv preprint arXiv: 2305.14507*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Mal-

colm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux,

Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Wojciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, Xiang-Hai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeck-

emeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv: 2403.05530*.

Maria Tigina, Anastasiia Birillo, Yaroslav Golubev, H. Keuning, Nikolay Vyahhi, and T. Bryksin. 2023. Analyzing the quality of submissions in online programming courses. *IEEE/ACM 45th International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv: 2307.09288*.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13894–13908, Bangkok, Thailand. Association for Computational Linguistics.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. *arXiv preprint arXiv: 2402.14804*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv: 2409.12191*.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. 2024c. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. 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 2174–2199, Mexico City, Mexico. Association for Computational Linguistics.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024d. Large language models for education: A survey and outlook. *arXiv preprint arXiv: 2403.18105*.

Shen Wang, Tianlong Xu, Hang Li, Chaoli Zhang, Joleen Liang, Jiliang Tang, Philip S. Yu, and Qingsong Wen. 2024e. Large language models for education: A survey and outlook. *arXiv preprint arXiv: 2403.18105*.

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

Hou Xinyi, Zhao Yanjie, Liu Yue, Yang Zhou, Wang Kailong, Li Li, Luo Xiapu, Lo David, Grundy John, and Wang Haoyu. 2023. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, Aoxiao Zhong, Kun Wang, Hui Xiong, Philip S. Yu, Xuming Hu, and Qingsong Wen. 2024. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv: 2410.04509*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv: 2407.10671*.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024a. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *CVPR*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2024b. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv: 2409.02813*.

Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. 2024. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wenxuan Zhang, Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Advances in Neural Information Processing Systems*, 36:5484–5505.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv: 2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2023. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv: 2304.06364*.

Zihao Zhou, Shudong Liu, Maizhen Ning, Wei Liu, Jindong Wang, Derek F. Wong, Xiaowei Huang, Qiufeng Wang, and Kaizhu Huang. 2024. Is your model really a good math reasoner? evaluating mathematical reasoning with checklist. *arXiv preprint arXiv: 2407.08733*.

Appendix

A Additional details of FERMAT

A.1 Distribution of Math Domains and Perturbation Domains in FERMAT

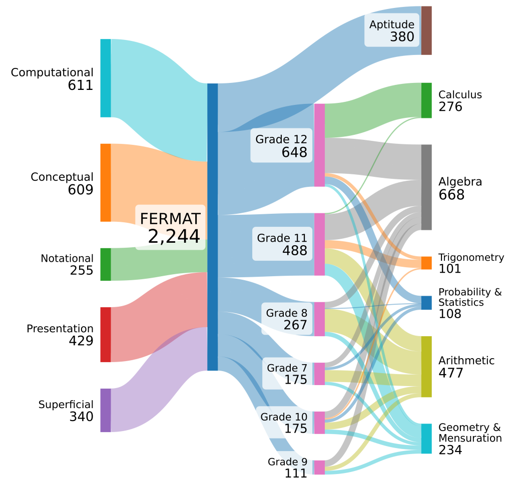

Figure 5: Distribution of different error types (left) across educational levels (middle) and math topics (right) within FERMAT.

Domain	Subdomains
Arithmetic	Decimals, Exponents, Factorization, Fractions, Percentages, Proportion, Ratio, Squares, Cubes, Arithmetic Progression, Permutation, Combination, Sequences
Algebra	Complex Numbers, Determinants, Expressions, Linear Equations, Linear Inequalities, Matrices, Polynomial, Relations, Functions, Sets, Vectors
Mensuration & Geometry	3D Geometry, Circles, Ellipse, Hyperbola, Lines, Parabola, Perimeter, Polygon, Surface Area, Triangles, Volume
Calculus	Continuity, Definite Integral, Derivatives, Differential Equations, Differentiability, Indefinite Integral, Limits, Maxima Minima, Area Under Curve
Probability & Statistics	Bayes Theorem, Conditional Probability, Data Handling, Independent Events
Trigonometry	Inverse Trigonometric Equations, Trigonometric Functions
Aptitude	Quantitative Aptitude

Table 5: Domains and Subdomains in FERMAT

A.2 Human Verification of Perturbations

We enlisted three mathematically proficient graduate students familiar with VLMs to verify the perturbations. Each annotator received task instructions, the original question-answer pair, the perturbation category, and the GPT-4o-generated perturbed pair. The annotators categorized each perturbation as either: (1) Valid Perturbation, (2) Invalid Perturbation, or (3) Not Relevant. Detailed guidelines explaining the expected perturbations and the rationale for their validity were provided. To assist in this task, we developed a custom application, shown in Figures 6 and 7. The interface enables side-by-side comparison of original and perturbed answers to facilitate accurate categorization.

Perturbations were classified as "Valid" only if they conformed to the specified perturbation category. Those irrelevant to the category or of insufficient quality were labeled as "Invalid". Those that had minor mistakes were classified as "Not Relevant" and subsequently were resurrected after minor adjustments.

A.3 Handwritten transcription

We engaged 43 experienced OCR annotators to manually generate perturbed question-answer pairs, using diverse writing instruments, paper types, lighting conditions, and paper qualities. Annotators reproduced the GPT-4o generated perturbed question-answer pairs verbatim, captured photographs, and uploaded the images directly. A dedicated application was developed to streamline this process, as illustrated in Figures 8 and 9.

B Manual annotation quality assessment

Three graduate students with expertise in Vision Language Models reviewed the annotations for quality assurance. Each reviewer received task instructions, the original question-answer pair, the perturbation reasoning and category, the GPT-4o-generated perturbed pair, and its handwritten version. Annotations were classified as: (1) High-Quality, (2) Low-Quality, or (3) Not Sure. The application interface used for this task is depicted in Figures 10, 11, and 12.

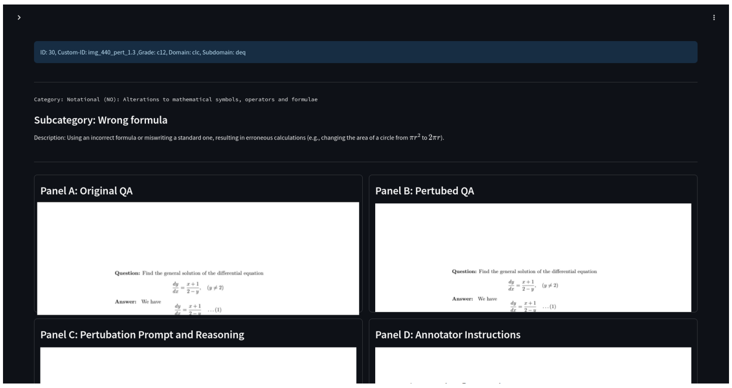

Figure 6: Interface for manual verification of perturbations (1).

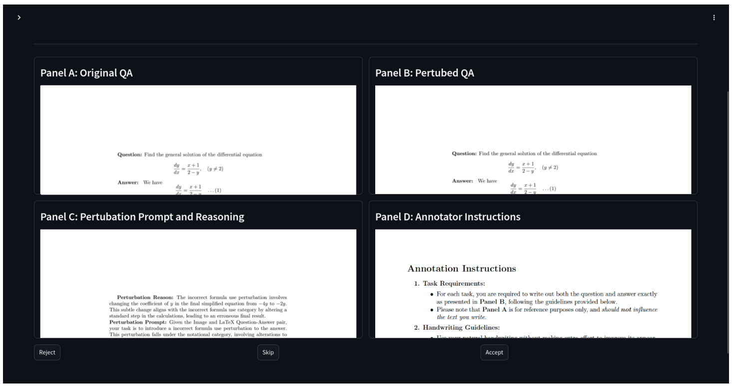

Figure 7: Interface for manual verification of perturbations (2).

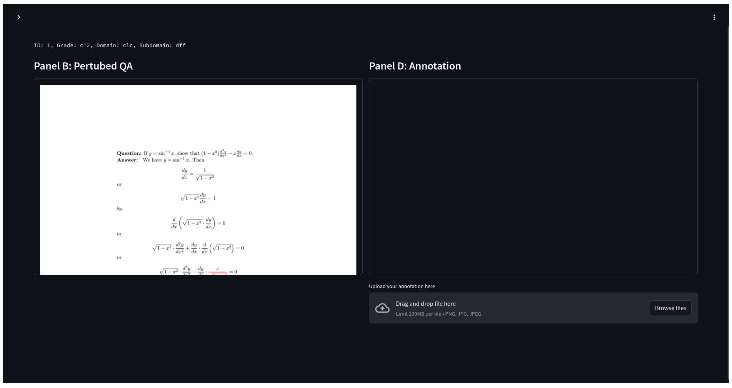

Figure 8: Interface for annotators to upload handwritten perturbed question-answer pairs (1).

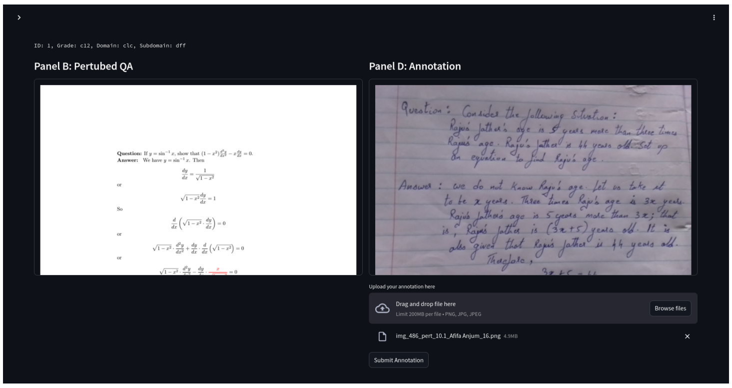

Figure 9: Interface for annotators to upload handwritten perturbed question-answer pairs (2).

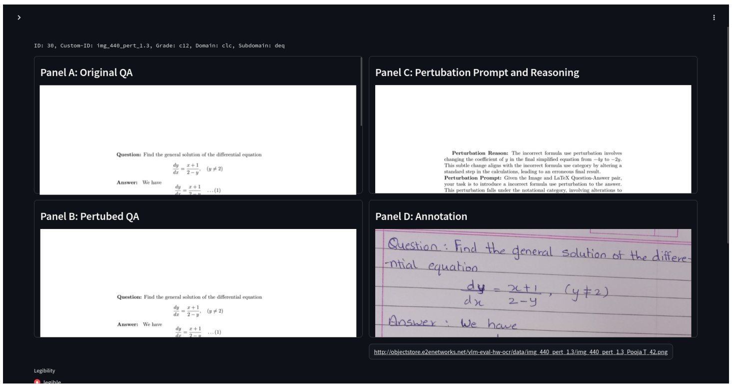

Figure 10: Interface for annotation quality assessment (1).

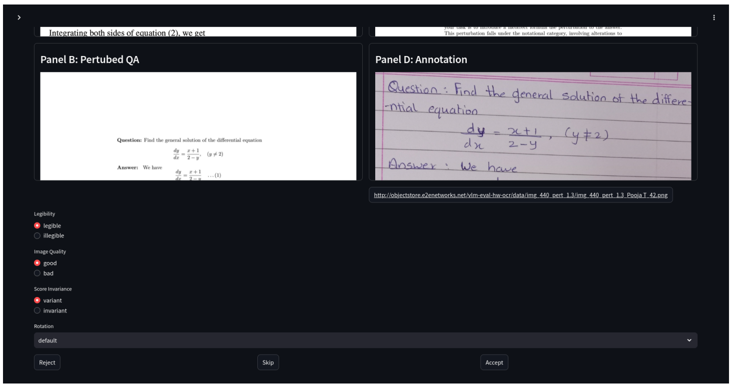

Figure 11: Interface for annotation quality assessment (2).

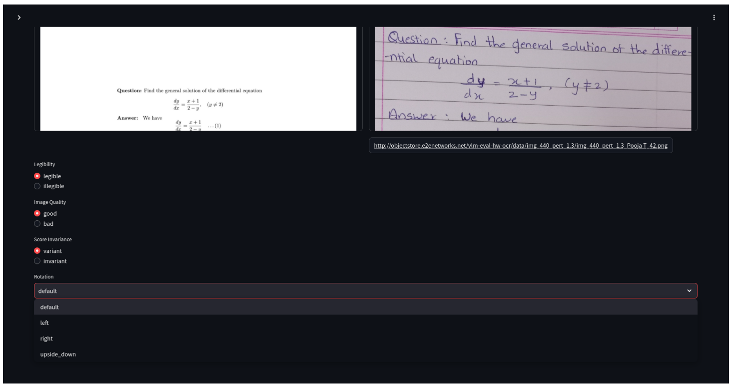

Figure 12: Interface for annotation quality assessment (3).

Each annotation was further evaluated based on the following criteria: legibility (legible or illegible), image quality (good or bad), score invariance (variant or invariant), and rotation (default, left, right, or upside down). This rigorous process ensures adherence to perturbation categories and accurate identification of score invariance, critical for benchmark quality.

C LLMs as Evaluators

C.1 Evaluator Details and Prompt Design

Using human evaluation to assess VLM localization and correction outputs for FERMAT samples is both cost-intensive and laborious. Furthermore, this process must be repeated with the emergence of each new state-of-the-art VLM, limiting scalability and rapid adoption. To address these challenges, we employ LLMs as verifiers of VLM outputs, specifically leveraging GPT-4o. This decision is based on GPT-4o's broad adoption and strong performance on reasoning-based tasks.

VLMs	LLM Accuracy
GPT-4o	0.96
Llama-3.2-11B	0.94
Pixtral-12B	0.91
Phi-3.5-VI	0.94
Overall	0.94

Table 6: Comparison of GPT-4o performance with respect to human evaluation in verifying the correctness of error localization outputs across various VLMs. Higher values indicate better performance.

We assessed the reliability of GPT-4o as an Evaluator LLM through a controlled study involving 464 randomly selected outputs from **EL** across four VLMs: GPT-4o, Llama-3.2-11B, Pixtral-12B, and Phi-3.5-VI. Graduate students independently evaluated the correctness of the VLMs' error localization outputs. These outputs were then provided to GPT-4o along with detailed prompts (Figures 29, 30) outlining the scoring criteria, including explicit guidelines on awarding or withholding scores. For error localization, we prompt the LLM, denoted as $g(\cdot)$, using the VLM's output $text_{loc}$ (§4.2) as the predicted text, alongside the perturbed answer (A_{pert}) and the explanation for the perturbation (exp) (§3.3) as the ground truth. The LLM is tasked with determining whether the VLM correctly localizes the error(s). This can be formally represented as $g(text_{loc}, A_{pert}, exp_{pert}) \rightarrow (reason, True/False)$. Similarly, for error correction, we prompt the LLM, $g(\cdot)$, using the VLM's corrected output (A_{corr}) (§4.3) as the predicted solution and the original solution (A_{gold}) as the ground truth. The LLM is asked to verify if the VLM accurately corrected the solution. This process is represented as $g(A_{corr}, A_{gold}) \rightarrow (reason, True/False)$. Prompts are designed to cover potential output scenarios and includes comprehensive guidelines to ensure consistent scoring.

Our findings indicate that GPT-4o achieves 94% accuracy in aligning with human judgments of localization correctness. Table 6 presents a comparison of GPT-4o's performance with human evaluation, demonstrating its effectiveness as an Evaluator LLM.

C.2 Testing the reliability of Evaluator LLM

We developed a dashboard to compare human and LLM performance in reasoning and decision-making. The evaluation was based on 464 randomly sampled items from the dataset, ensuring equal representation across all perturbation categories. The evaluation compared LLM reasoning with human reasoning, LLM decisions with human decisions, and LLM decisions with its own reasoning. This analysis is crucial to determine whether LLMs can effectively replace human annotators in error localization and correction tasks.

D VLM Performance in Error Detection

Models	Non-cascaded			
	ED		ED+OCR	
	ACC	F1	ACC	F1
Gemini-1.5-Flash	0.51	0.55	0.67	0.70
Gemini-1.5-Pro	0.54	0.58	0.68	0.71
GPT-4o	0.51	0.55	0.59	0.63
GPT-4o-mini	**0.78**	**0.74**	0.73	0.72
Llama-3.2-11B	0.68	0.68	0.72	0.70
Llama-3.2-90B	0.66	0.67	0.70	0.72
Pixtral-12B	0.77	0.72	**0.75**	**0.73**
Pixtral-124B	0.30	0.27	0.61	0.65
Phi-3.5-VI	0.70	0.70	0.59	0.62

Table 7: Performance of VLMs on error detection task with accuracy (**Acc**) and **F1** scores as the evaluation metrics. Higher values indicate better performance.

We provide the Accuracy and F1 scores for the Error Detection task across all nine VLMs in Table 7. Interestingly, for Error Detection (**ED**), GPT-4o, Gemini-1.5-Pro and Gemini-1.5-Flash models perform only slightly better than random, while GPT-4o-mini outperforms all other models, a behavior that is significantly different from their performance in error localization and correction while looking at Accuracy and F1 score as a metric. To further investigate this, we analyze the explanations (exp) generated as part of the **ED** task output for all models to determine if they correctly identify errors. As shown in Figure 13, we observe that smaller models, including GPT-4o-mini, predict a high rate of positives with incorrect reasoning, indicating that these models incorrectly classify many instances as errors. Given the class imbalance in **Fermat**, this results in inflated Accuracy and F1 scores. In contrast, larger models such as GPT-4o and Gemini-1.5-Pro produce significantly fewer False Positives. This finding aligns with previous research by Li et al. (2024b), which demonstrated that such models are generally more cautious in error detection.

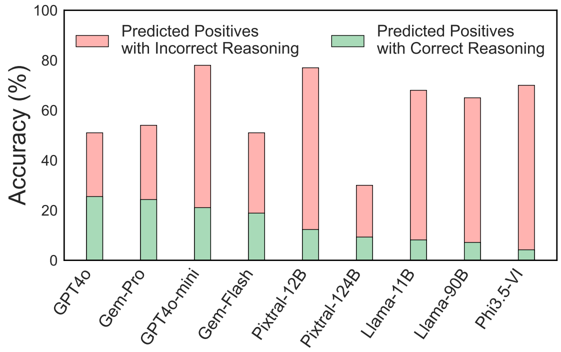

Figure 13: Performance of VLMs on the error detection task: comparing cases where predicted positives align with their reasoning against cases where they do not.

E Performance of VLMs in Cascaded Setup

We observe that bigger models like GPT-4O, GEMINI-1.5-PRO and GEMINI-1.5-FLASH perform worse in a cascaded Error Evaluation setup due to their cautious nature of identifying errors in a solution. On the other extreme, PIXTRAL-124B gets heavily penalized due to its very high false negative prediction rate, resulting in degraded error evaluation performance. Table 3 shows the modelwise performance on the cascaded setup. Sankey graphs illustrating the performance of VLMs in the Cascaded Setup, along with their intermediate output values, are shown in Figure 3 and detailed further in Figure 14 through Figure 22.

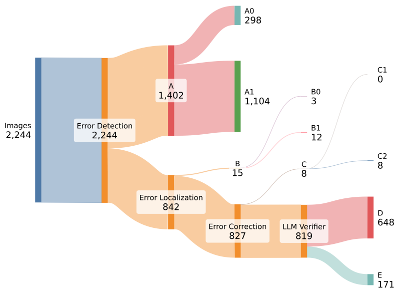

Figure 14: Breakdown of intermediate and final output proportions in GPT4o.

F Prompts used for various Experiments

The task-specific evaluation prompts for all Vision-Language Models (VLMs) assessed on FERMAT are detailed below in Figure 23 through Figure 32. For each task, we ensured consistent evaluation by using identical prompts across all models

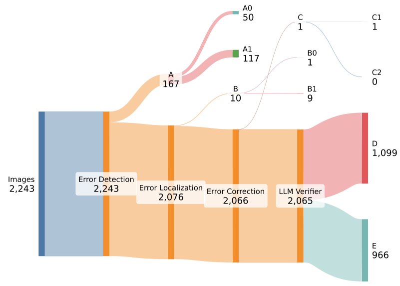

Figure 15: Breakdown of intermediate and final output proportions in GPT4o-mini.

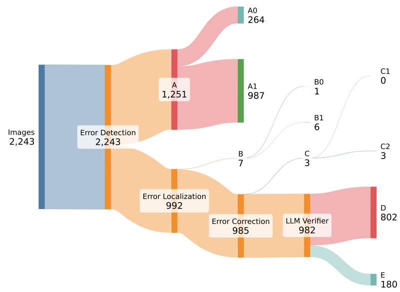

Figure 16: Breakdown of intermediate and final output proportions in Gemini Pro.

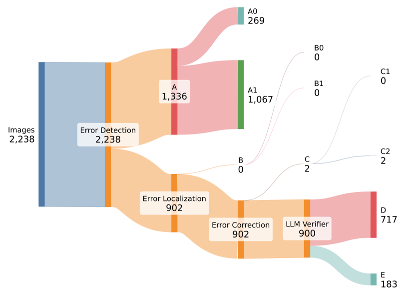

Figure 17: Breakdown of intermediate and final output proportions in Gemini Flash.

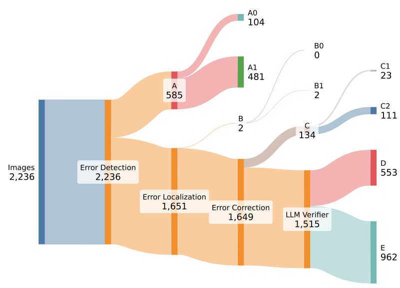

Figure 18: Breakdown of intermediate and final output proportions in LLaMA Large.

and setting the sampling temperature to zero to ensure reproducibility. Similarly, for the Evaluator LLM, we employed GPT-4O with a temperature

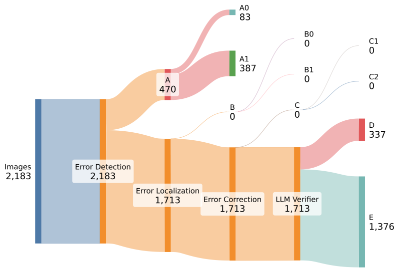

Figure 19: Breakdown of intermediate and final output proportions in LLaMA.

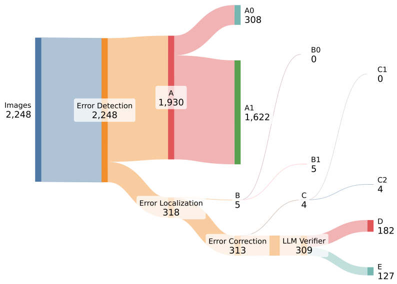

Figure 20: Breakdown of intermediate and final output proportions in Pixtral Large.

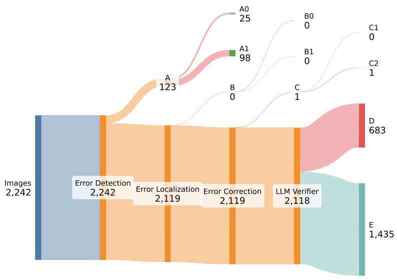

Figure 21: Breakdown of intermediate and final output proportions in Pixtral.

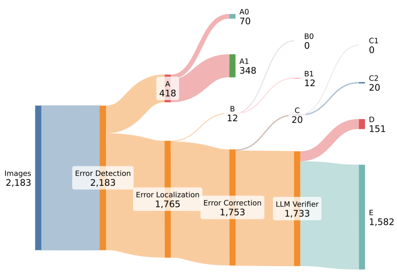

Figure 22: Breakdown of intermediate and final output proportions in Phi.

```
The image provided contains a handwritten problem with both a
Question and an Answer at a middle or high school level. Your
task is to explicitly perform OCR on the handwritten text and
extract the content in LaTeX format. Return only the extracted
content exactly as it appears in the image, formatted in
LaTeX.

                Ensure that no extra information is added that
is not in the image.

                Please return the LaTeX output as follows:
                **Question:**<Extracted Question text in
LaTeX>
                **Answer:**<Extracted Answer text in LaTeX>
```

Figure 23: Prompt for OCR Extraction from Image.

```
The image provided contains a handwritten math problem consisting
of both a Question and an Answer at a middle or high school level.
Your task is to analyze the Answer to determine whether there is
any error. Begin by providing a brief reasoning for your analysis,
explaining where and why you believe an error is present or absent
in the Answer. If the problem is multiple-choice (MCQ), judge the
presence or absence of error based only on the explanation given in
the Answer, not the option selected by the student.

                After the reasoning, provide a binary output
indicating whether an error exists (1 for error, 0 for no error).

                Please follow the exact format below without adding
any extra information:

                **Reasoning:** <Brief Explanation of Error Presence
or Absence>
                **Error:** <0 or 1>
```

Figure 24: Prompt for Error Detection.

of zero.

```
The image provided contains a handwritten math problem
with both a Question and an Answer at a middle or high
school level. Your task is to analyze the Answer,
identify any errors, and if present, localize the errors
in the **Error Localization:** field below.

        Begin by providing a brief reasoning for your
analysis, explaining where and why an error is present
or absent in the Answer. If the problem is
multiple-choice (MCQ), focus on the explanation in the
Answer and not the option selected by the student when
identifying errors.

        After the reasoning, based on your analysis,
localize the exact lines or steps in the Answer where
the error occurs, in the **Error Localization** field.
If no error is present, mention 'NA' in the **Error
Localization:** field.

        Please follow the exact format below without
adding any extra information:

        **Reasoning:** <Brief Explanation of Error
Presence or Absence>
        **Error Localization:** <Specific lines or steps
in the Answer where the error occurs, or 'NA' if no
error>
```

Figure 25: Prompt for Error Localization.

```
The image provided contains a handwritten math problem
with both a Question and an Answer at a middle or high
school level. Your task is to analyze the Answer,
identify any errors, and if present, correct the errors
in the Answer and return it in the **Corrected Answer
LaTeX:** field below.

        Begin by providing a brief explanation of where
and why an error is present or absent in the Answer. If
the problem is multiple-choice (MCQ), focus on the
explanation provided in the Answer rather than the
option selected by the student when identifying errors.

        Next, based on your analysis, give the correct
Answer in LaTeX format and ensure that the LaTeX Answer
is meaningful, logical, and aligns with the instructions
in the Question. If the problem is multiple-choice
(MCQ), return the full LaTeX Answer with the complete
corrected explanation retained as visible in the image,
along with the correct option that should have selected.
If no error is present in the Answer, mark 'NA' in the
**Corrected Answer LaTeX:** field.

        Please follow the exact format below without
adding any extra information:

        **Reasoning:** <Brief Explanation of Error
Presence or Absence>
        **Corrected Answer LaTeX:** <Complete Corrected
LaTeX Answer, or 'NA' if no error>
```

Figure 26: Prompt for Error Correction.

```
The image provided contains a handwritten math problem, containing both a Question and its
Answer at a middle or high school level. The Answer contains one or multiple instances of
errors. Your task is to analyze the Answer, identify all instances of errors, and localize these
errors in the **Error Localization** field below.

        - Begin by providing a brief reasoning for your analysis, clearly explaining the nature
and location of the error in the Answer. If the problem is multiple-choice (MCQ), focus on the
explanation within the Answer or the selected option when identifying errors.
        - Based on your reasoning, pinpoint the exact lines or steps in the Answer where the
error occurs and include them in the **Error Localization** field.
        - If no error can be confidently identified, mention 'NA' in the **Error Localization**
field.

        **Format:**
        Please strictly adhere to the format below without adding any additional information:

        **Reasoning:** <Brief explanation of the error, its nature, and why it is incorrect>
        **Error Localization:** <Specific lines or steps where the error occurs, or 'NA' if no
error>
```

Figure 27: Prompt for Cascaded Error Localization.

```
The image provided contains a math problem, including both a Question and its Answer at a middle or high school level.
The Answer contains one or more instances of errors in the Answer, localized below:

        [INSERT ERROR LOCATION FROM PREVIOUS STEP]

        Your task is to analyze the Answer and, based on the localized errors provided, correct the errors in the Answer
and return it in the **Corrected Answer LaTeX** field below.

        - Start by briefly explaining whether you agree with the identified errors and why. If the problem is
multiple-choice (MCQ), focus on the explanation provided in the Answer and the option selected by the student when
identifying errors.
        - Next, based on your analysis, give the corrected Answer in LaTeX format. Ensure that the LaTeX Answer is
meaningful, logical, and aligns with the instructions in the Question.
        - If the problem is multiple-choice (MCQ), return the complete LaTeX Answer, including the corrected explanation
visible in the image, along with the correct option that should have been selected.
        - If no error can be confidently identified, mention 'NA' in the **Corrected Answer LaTeX** field.

        Please strictly adhere to the format below without adding any extra information:

        **Reasoning:** <Brief explanation of the error, its nature, and why it is incorrect or absent>
        **Corrected Answer LaTeX:** <Complete corrected LaTeX Answer, or 'NA' if no error to correct>
```

Figure 28: Prompt for Cascaded Error Correction.

```
You are tasked with evaluating a model's performance in identifying and
locating errors in a math problem solution. Your goal is to assess
whether the **Predicted Error Location** sufficiently matches the
**Ground Truth (GT) Error Location**, considering both the predicted
location and the model’s reasoning.

            **Provided Information:**
            - **Original Question**: The math problem being solved.
            - **Answer**: The solution to the above math problem with no
errors.
            - **GT Error Location**: There is no error in the solution.
            - **Predicted Error Location**: Error Location(s) as
identified by the model.
            - **Reasoning behind Predicted Error Location**: The model’s
explanation for its prediction.

            **Goal:**
            To evaluate whether the model was able to output that there
was no error in **Answer** by mentioning 'NA' (Not Applicable) explicitly
or a similar statement indicating that there are no errors to localize.

            **Your Task:**

            1. **Evaluate Prediction and Reasoning**:
            - Compare whether the **Predicted Error Location** aligns
with the **GT Error Location** by either explicitly stating 'NA' or
mentioning indicating that there are no errors in the Answer.
            - Examine the **Reasoning behind Predicted Error Location**
to determine whether it justifies the predicted location. If **Predicted
Error Location** is marked 'NA', assess whether the reasoning provides
sufficient evidence to verify why there are no errors in the solution.

            2. **Scoring Guidelines**:
            - NOTE: The **Answer** contains no errors.
            - Award **1** in the following cases:
                - The **Predicted Error Location**, combined with the
**Reasoning behind Predicted Error Location**, mentions 'NA' (Not
Applicable) explicitly or a similar statement indicating that there are
no errors to localize.
                - The **Reasoning behind Predicted Error Location**
sufficiently justifies the prediction when **Predicted Error Location**
is marked 'NA'
            - Award **0** in the following cases:
                - **Predicted Error Location** tries to localize some
error within the answer wrongly.
                - The reasoning fails to justify the prediction,
particularly when **Predicted Error Location** is 'NA.'

            **Final Verdict**:
            - If the **Predicted Error Location** mentions 'NA'
explicitly or mentions that there are no errors in the answer, award a
score of **1** to it for correctness, else if it tries to localize some
error within the answer, penalize it by awarding a score of **0**.

            **Response Format:**
                **Reasoning:** <Clear explanation assessing the
prediction and reasoning, referencing key details where relevant.>
                **Is Error Location Correct:** <0 or 1>

            **Example:**
            - If **Predicted Error Location** mentions 'NA' or 'The
solution is error-free' or something similar, score **1**.
            - If **Predicted Error Location** tries to localize some
error in the solution, score **0**.
```

Figure 29: Prompt for LLM Verification of Error Localization outputs from VLM when the Problem Solution is error-free.

```
You are tasked with evaluating a model's performance in identifying and locating errors in
a math problem solution. Your goal is to assess whether the **Predicted Error Location**
sufficiently matches the **Ground Truth (GT) Error Location**, considering both the
predicted location and the model’s reasoning.

            **Provided Information:**
            - **Original Question**: The math problem being solved.
            - **Answer**: The solution to the above problem containing errors.
            - **GT Error Location**: The true location(s) of error(s) in the solution.
            - **Predicted Error Location**: Error Location(s) as identified by the model.
            - **Reasoning behind Predicted Error Location**: The model’s explanation for
its prediction.

            **Goal:**
            To evaluate whether the model was able to output the locations of all errors
in **Answer** either through **Predicted Error Location** or **Reasoning behind Predicted
Error Location**.

            **Your Task:**

            1. **Analyze Errors**: Review the **Answer** and the **GT Error Location** to
understand the error(s) location(s).

            2. **Evaluate Prediction and Reasoning**:
            - Compare the **Predicted Error Location** to the **GT Error Location** for
alignment.
            - Examine the **Reasoning behind Predicted Error Location** to determine
whether it justifies the predicted location. If **Predicted Error Location** is explicitly
marked 'NA' or mentions that there are no errors in **Answer**, assess whether the
reasoning provides sufficient evidence to identify the true error(s).

            3. **Scoring Guidelines**:
            - NOTE: The **Answer** contains one or more errors with the error location
mentioned in **GT Error Location**.
            - Award **1** in the following cases:
                - The **Predicted Error Location**, combined with the reasoning correctly
aligns with the **GT Error Location**, capturing all key error instances.
                - If **Predicted Error Location** is marked 'NA' or mentions that there is
no errors in the answer, look at the **Reasoning behind Predicted Error Location** and if
it sufficiently localizes errors, award **1**.
            - Award **0** in the following cases:
                - Key errors in the **GT Error Location** are missed by both the
prediction and the reasoning.
                - Errors not present in the **GT Error Location** are incorrectly
identified in the **Predicted Error Location**
                - The reasoning also fails to locate the errors, particularly when
**Predicted Error Location** is 'NA.'

            **Final Verdict**:
            - Prioritize **Reasoning** if **Predicted Error Location** is marked 'NA'
            - Balance your assessment between the **Predicted Error Location** and the
**Reasoning** for a comprehensive evaluation. If the **Predicted Error Location** contains
'NA', consider the **Reasoning behind Predicted Error Location** for evaluating
correctness with respect to **GT Error Location**.

            **Response Format:**
                **Reasoning:** <Clear explanation assessing the prediction and reasoning,
referencing key details where relevant.>
                **Is Error Location Correct:** <0 or 1>

            **Example:**
            - If **Predicted Error Location** misses a key error but the reasoning clearly
identifies and justifies it, score **1** based on combined performance.
            - If **Predicted Error Location** and **Reasoning behind Predicted Error
Location** both fail to identify the error locations properly, score **0**.
```

Figure 30: Prompt for LLM Verification of Error Localization outputs from VLM when the Problem Solution contains errors.

```
You are tasked with evaluating a model's performance by comparing two answers for
a math problem to assess how accurately the model corrected the response.

            **Provided Information:**
            - **Original Question**: The math problem being solved.
            - **Original Answer**: The ground-truth correct answer.
            - **Erroneous Answer**: The answer with superficial modifications to
the **Original Answer**. It contains no errors. This has been provided as input to
the model.
            - **Corrected Answer**: The answer as corrected by the model.
            - **Reasoning behind Corrected Answer**: The model's explanation
justifying its corrections or changes.

            **Goal:**
            To evaluate whether the model is able to determine that the
**Corrected Answer** should be 'NA' (Not Applicable) or a similar statement
indicating that there are no errors to correct.


            **Your Task:**

            1. Determine whether the **Corrected Answer** either explicitly states
'NA' or mentions that there are no errors to correct in the Answer.


            2. **Scoring Guidelines**:

            - Award **1** in the following cases:
            - The **Corrected Answer** mentions 'NA', indicating there was no
error to correct.
            - The **Corrected Answer** mentions that there are no errors in the
answer or similar statements.
            - The **Corrected Answer** essentially restates the **Erroneous
Answer** without changes, indicating that the model has returned back the answer
due to absence of errors. As long as the corrected answer is essentially identical
(or very close) to the **Erroneous Answer**, treat that as if the model recognized
no errors needed correction.
            - there are minor stylistic changes (e.g., rephrasing sentences,
changing formatting) in **Corrected Answer** but does not introduce any
mathematical or logical errors

            - Award **0** in the following cases:
            - The model introduces new significant errors in the **Corrected
Answer** that are not present in the **Original Answer**.
            - The model fails to identify that the **Corrected Answer** should be
'NA' or mention that there are no errors in the answer.


            **Final Verdict**:
            - Determine if the **Corrected Answer** either explicitly states 'NA',
or mentions that there are no errors to correct, or provides a **Corrected
Answer** text that closely matches the **Original Answer** with minimal
differences. In all these cases, award **1**, else **0**.


            **Response Format:**
            - **Reasoning:** <Brief explanation of whether the Corrected Answer is
an accurate match to the Original Answer.>
            - **Is Error Correction Accurate:** <0 or 1>

            **Example:**
            - Score **1** if the **Corrected Answer** mentions 'NA', indicating
there was no error to correct.
            - Score **0** if the **Corrected Answer** introduces new errors not
present in the **Original Answer**.
```

Figure 31: Prompt for LLM Verification of Error Correction outputs from VLM when the Problem Solution is error-free.

```
You are tasked with evaluating a model's performance by comparing two answers for a math
problem to assess how accurately the model corrected the response.

            **Provided Information:**
            - **Original Question**: The math problem being solved.
            - **Original Answer**: The ground-truth correct answer.
            - **Erroneous Answer**: The answer with errors that need correction. This has
been provided as input to the model.
            - **Corrected Answer**: The answer as corrected by the model.
            - **Reasoning behind Corrected Answer**: The model's explanation justifying
its corrections or changes.

            **Goal:**
            To evaluate whether the **Corrected Answer** as corrected by the model is a
logically consistent and close match to the **Original Answer**.


            **Your task:**

            1. Determine whether the **Corrected Answer** properly corrects the error(s)
in **Erroneous Answer**. Next, check whether **Corrected Answer** closely matches the
**Original Answer**. The **Corrected Answer** should not contain any new significant
errors that were not present in the **Original Answer**.
            2. If the problem is multiple-choice (MCQ), consider the correction accurate
only if the selected option in **Corrected Answer**  exactly matches the **Original
Answer**, and the answer explanation in the **Corrected Answer** logically matches the
explanation in the **Original Answer**. Focus on the explanation to assess correctness,
rather than the just the option selected.

            3. **Scoring Guidelines**:
            - **Almost exact match**: The **Corrected Answer** does not need to be an
exact match but should align closely in content with the **Original Answer**.

            - Award **1** in the following cases:
            - All errors in **Erroneous Answer** are rectified in **Corrected Answer** and
the **Corrected Answer** is a logical, corrected match to the **Original Answer**.
            - If the **Corrected Answer** is marked 'NA' or mentions that there are no
errors in the answer, look at the **Reasoning behind Corrected Answer** and if it
sufficiently points out the errors in **Erroneous Answer** and explains their corrected
versions, award **1**.
            - If the **Corrected Answer** is logically correct and fixes the error(s), but
uses a different valid approach or representation than the **Original Answer**, still
award **1**.

            - Award **0** in the following cases:
            - If there are still  errors remaining in the **Corrected Answer** that are
not present in the **Original Answer**, i.e. all mistakes in the **Erroneous Answer** are
not corrected properly.
            - If **Corrected Answer** is 'NA' or mentions that there are no errors in the
answer, look at the **Reasoning behind Corrected Answer**. If that also fails to
sufficiently point out the errors in **Erroneous Answer** and explains their corrected
versions, award **0**.


            **Final Verdict**:
            - Prioritize **Reasoning behind Corrected Answer** to assess correctness if
**Corrected Answer** is marked 'NA'. If reasoning also fails to identify the error
locations in **Erroneous Answer** and mention their proper corrections, only then award
**0**, else award **1**.
            - Determine whether the **Corrected Answer** closely matches the **Original
Answer** in logical correctness, with minimal differences.


            **Response Format:**
            **Reasoning:** <Brief explanation of whether the Corrected Answer is an
accurate match to the Original Answer>
            **Is Error Correction Accurate:** <0 or 1>

            **Example:**
            - If **Corrected Answer** misses a key error in **Erroneous Answer**, but the
reasoning clearly identifies and justifies it, score **1** based on combined performance.
            - If **Corrected Answer** and **Reasoning behind Correction** both fail to
identify the error locations and mention their proper corrections, score **0**.
```

Figure 32: Prompt for LLM Verification of Error Correction outputs from VLM when the Problem Solution contains errors.

```
You are provided with a handwritten math problem
image, containing a Question and its corresponding
Answer. This problem is from the **[INSERT
SUBDOMAIN]** subdomain in **[INSERT DOMAIN]** and is
designed for **[INSERT GRADE INFO]** math students.
Analyze the Answer for accuracy by following these
steps:

        1. Evaluate for Errors: Examine the solution
process in the Answer. If an error is present,
describe where it occurs and why it's incorrect based
on mathematical reasoning, logical progression, or
calculation accuracy. If no error is found, state why
the solution is correct and aligns with the problem
requirements.

        2. For Multiple-Choice (MCQ) Problems: Assess
the correctness of both the option chosen and the
explanation given, ensuring they are consistent and
valid.

        3. Binary Decision: After providing your
reasoning, indicate whether an error exists. Use "1"
to represent an error and "0" to indicate no error.

        Please follow the exact format below without
adding any extra information:

        **Reasoning:** <Brief Explanation of Error
Presence or Absence>
        **Error:** <0 or 1>
```

Figure 33: Level 1 Error Detection Prompt for Section 5.3.

```
You are provided with a handwritten math problem image,
containing a Question and its corresponding Answer. This problem
is from the **[INSERT SUBDOMAIN]** subdomain in **[INSERT
DOMAIN]** and is designed for **[INSERT GRADE INFO]** math
students.

        The problem solution may contain one or more errors from
the following categories:
        [INSERT PERTURBATION CATEGORIES]


        Analyze the Answer for correctness by following these
steps:

        1. Evaluate for Errors: Examine the solution process in
the Answer. If an error is present, describe where it occurs and
why it's incorrect based on mathematical reasoning, logical
progression, or calculation accuracy. If no error is found, state
why the solution is correct and aligns with the problem
requirements.

        2. For Multiple-Choice (MCQ) Problems: Assess the
correctness of both the option chosen and the explanation given,
ensuring they are consistent and valid.

        3. Binary Decision: After providing your reasoning,
indicate whether an error exists. Use "1" to represent an error
and "0" to indicate no error.

        Please follow the exact format below without adding any
extra information:

        **Reasoning:** <Brief Explanation of Error Presence or
Absence>
        **Error:** <0 or 1>
```

Figure 34: Level 2 Error Detection Prompt for Section 5.3.

```
You are provided with a handwritten math problem image, containing a
Question and its corresponding Answer. This problem is from the
**[INSERT SUBDOMAIN]** subdomain in **[INSERT DOMAIN]** and is designed
for **[INSERT GRADE INFO]** math students.

        There may be a [INSERT PERTURBATION SUPERCATEGORY] error
present in the solution such as [INSERT PERTURBATION CATEGORIES].

    More specifically, the error could be [INSERT SPECIFIC
PERTURBATION],i.e. [INSERT PERTURBATION DESCRIPTION]

    Examples of this type of error include:
    [INSERT PERTURBATION EXAMPLES]


        Analyze the Answer for correctness by following these steps:

        1. Evaluate for Errors: Examine the solution process in the
Answer. If an error is present, describe where it occurs and why it's
incorrect based on mathematical reasoning, logical progression, or
calculation accuracy. If no error is found, state why the solution is
correct and aligns with the problem requirements.

        2. For Multiple-Choice (MCQ) Problems: Assess the correctness
of both the option chosen and the explanation given, ensuring they are
consistent and valid.

        3. Binary Decision: After providing your reasoning, indicate
whether an error exists. Use "1" to represent an error and "0" to
indicate no error.

        Please follow the exact format below without adding any extra
information:

        **Reasoning:** <Brief Explanation of Error Presence or Absence>
        **Error:** <0 or 1>
```

Figure 35: Level 3 Error Detection Prompt for Section 5.3.

```
You are provided with a math problem, containing a Question and its corresponding
Answer. This problem is from the **[INSERT SUBDOMAIN]** subdomain in **[INSERT DOMAIN]**
and is designed for **[INSERT GRADE INFO]** math students.

    There may be a [INSERT PERTURBATION SUPERCATEGORY] error present in the solution
such as [INSERT PERTURBATION CATEGORIES].

    More specifically, the error could be [INSERT SPECIFIC PERTURBATION],i.e. [INSERT
PERTURBATION DESCRIPTION]

    Examples of this type of error include:
    [INSERT PERTURBATION EXAMPLES]

    For reference, here is a problem with a similar error in the Answer:
    **Sample Question**: [INSERT ORIGINAL QUESTION]
    **Sample Answer with Error**: [INSERT ERRONEOUS ANSWER]
    **Error Location**: [INSERT ERROR LOCATION(S) IN ANSWER]



    Now, given the reference problem above, analyze the below Answer for correctness by
following these steps:

    1. Evaluate for Errors: Examine the solution process in the Answer. If an error is
present, describe where it occurs and why it's incorrect based on mathematical
reasoning, logical progression, or calculation accuracy. If no error is found, state why
the solution is correct and aligns with the problem requirements.

    2. For Multiple-Choice (MCQ) Problems: Assess the correctness of both the option
chosen and the explanation given, ensuring they are consistent and valid.

    3. Binary Decision: After providing your reasoning, indicate whether an error
exists. Use "1" to represent an error and "0" to indicate no error.

    Please follow the exact format below without adding any extra information:

    **Reasoning:** <Brief Explanation of Error Presence or Absence>
    **Error:** <0 or 1>
```

Figure 36: Level 4 Error Detection Prompt for Section 5.3.