ARB: ADVANCED REASONING BENCHMARK FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have demonstrated remarkable performance on various quantitative reasoning and knowledge benchmarks. However, many of these benchmarks are losing utility as LLMs get increasingly high scores, despite not yet reaching expert performance in these domains. We introduce ARB, a novel benchmark composed of advanced reasoning problems in multiple fields. ARB features problems in mathematics, physics, biology, chemistry, and law. As a subset of ARB, we introduce a challenging set of math and physics problems which require advanced symbolic reasoning and domain knowledge. We evaluate recent models such as GPT-4 and Claude on ARB and demonstrate that current models score well below 50% on more demanding tasks. In order to improve both automatic and assisted evaluation capabilities, we introduce a rubric-based evaluation approach, allowing GPT-4 to score its own intermediate reasoning steps. Further, we conduct a human evaluation of the symbolic subset of ARB, finding promising agreement between annotators and GPT-4 rubric evaluation scores.

1 INTRODUCTION

In recent years, models such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), PaLM (Chowdhery et al., 2022), and Chinchilla (Hoffmann et al., 2022) have shown increasing performance across a wide variety of natural language tasks ranging from translation to reasoning (Bubeck et al., 2023; Laskar et al., 2023). This rapid progress has been closely tracked and assessed by evaluating LLMs on benchmarks, which test model capabilities on a set of standardized problems. The GLUE benchmark (Wang et al., 2019b) for language understanding was first released in April 2018; but models such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019) in the following year were already powerful enough to necessitate the "SuperGLUE" benchmark (Wang et al., 2019a). Since then, the race between language models and benchmarks has increasingly favored the former.

Scaling up, model sizes and datasets alike, has led to rapid improvements on various natural language tasks on benchmarks like BIG-bench (Srivastava et al., 2022) and HELM (Liang et al., 2022). Neural scaling laws (Kaplan et al., 2020; Caballero et al., 2023; Alabdulmohsin et al., 2022) were used to predict the behavior of large scale models on various metrics. Nevertheless, LLM performance often increases unpredictably (Wei et al., 2022a), especially on tasks that require reasoning abilities. Predictions of performance on ML benchmarks often underestimate the rate of progress (Steinhardt, 2022). Since progress has been faster than anticipated, new benchmarks need to be more difficult.

Models such as ChatGPT have shown the ability to pass entry-level examinations in fields such as law (Bommarito II and Katz, 2022), medicine (Kung et al., 2023), economics (Caplan, 2023), and mathematics (Shakarian et al., 2023). Nevertheless, LLM understanding of many fields is reportedly shallow and unreliable (Shapira et al., 2023). *Expert reasoning* in domains with specialized knowledge is essential for automated systems to augment skilled professionals (Noy and Zhang, 2023).

In this paper, we introduce a new benchmark dataset, **ARB** (**A**dvanced **R**easoning **B**enchmark), designed to evaluate expert reasoning abilities in mathematics, physics, chemistry, biology, and law. To make the benchmark more challenging than previous benchmarks, we extract graduate-level tasks from resources intended for domain professionals. The mathematics and physics portions are more difficult than popular benchmarks such as MATH (Hendrycks et al., 2021), due to both the content and the question format. The performance of current models such as GPT-4 on the quantitative parts of ARB is very low using standard prompting methods.

Our dataset offers improvements over existing benchmarks:

- Hundreds of problems requiring expert reasoning in quantitative subjects, where LLMs are known to underperform;
- For mathematics and physics, all problems are short-answer and open-response questions, in contrast to the multiple-choice questions that dominated earlier benchmarks.

In addition, we propose an automated rubric-based method allowing self-evaluation of intermediate reasoning steps. While not currently a substitute for human evaluation, rubrics generated by GPT-4 have good coverage, and self-evaluation scores track human grading surprisingly well.

We provide the instructions to access the dataset in the supplementary material.

2 RELATED WORK

Improving the reasoning capabilities of LLMs had been a subject of recent interest, with a particular focus on advanced prompting techniques (Wei et al., 2022b; Kojima et al., 2023; Wang et al., 2023; Yao et al., 2023; Nye et al., 2021). Such techniques have seen increasingly successful applications in solving reasoning problems involving commonsense reasoning and mathematics, by promoting active reasoning processes within the LLMs before yielding final answers.

Model architectures such as Minerva (Lewkowycz et al., 2022) have exemplified the enhancement of reasoning capabilities through fine-tuning on extensive datasets covering math and reasoning tasks. This has yielded improved performance across several benchmarks, including MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and MMLU (Hendrycks et al., 2020). Concurrently, other lines of research (Li et al., 2023; Lightman et al., 2023; Cobbe et al., 2021) have investigated the application of verification techniques to augment and enhance LLM performance.

Most of the aforementioned work has typically evaluated techniques against math benchmarks (e.g., GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), SVAMP (Patel et al., 2021), ASDiv (Miao et al., 2020), AQuA (Ling et al., 2017), MAWPS (Koncel-Kedziorski et al., 2016), MultiArith (Roy and Roth, 2016)) and commonsense reasoning tasks (e.g., CSQA (Talmor et al., 2018), StrategyQA (Geva et al., 2021), HotpotQA (Yang et al., 2018)). Recently, several new benchmarks have been introduced for reasoning and planning tasks, such as the GPT-Planning Benchmark (Valmeekam et al., 2023), ALERT Reasoning Benchmark (Yu et al., 2022), and (Gendron et al., 2023). Additionally, comprehensive evaluation suites like the Chain-of-Thought Hub (Fu et al., 2023) have been proposed. Particularly related to our work is JEEBench (Arora et al., 2023)), which tests some of the same models as we do, on mathematics, physics and chemistry tasks. The main differences with our work are our quantitative problems are both somewhat harder and require deeper math/physics knowledge, and that their benchmark is entirely multiple-choice.

Most existing benchmarks are limited in difficulty and represent a restricted range of reasoning tasks. Moreover, recent advancements such as Minerva (Lewkowycz et al., 2022) have revealed that these benchmarks may not offer sufficient challenge. Of course, no single paper can solve these issues by itself; evaluation is co-evolving with capabilities and new benchmarks are always needed.

The rapid progress in LLM capabilities has led many to explore using LLMs in the LLM evaluation pipeline. Apart from using LLMs to generate evaluation tasks (Zhang et al., 2022; Perez et al., 2022), LLMs have increasingly been used as a proxy for human evaluation (Chiang and Lee, 2023; Liu et al., 2023; Fu et al., 2023; Kocmi and Federmann, 2023). Useful LLM-based evaluation for alignment has been done using rubrics (Bai et al., 2022). We explore the efficacy of rubrics for evaluation when applied to highly complex math and physics problems, proposing a path forward to issues discussed in Arora et al. (2023).

3 BENCHMARK

The key considerations when building an LLM benchmark are:

- **Difficulty.** Most tasks have to be out of reach of current models; a benchmark where many models score over 90% is not useful for tracking differential AI development.

- **Usefulness.** The tested skills should correlate with generally useful human skills.
- **Ease of evaluation.** It should be straightforward for the model creators to compare the performances of different models. The scores should be interpretable.
- **Minimizing data contamination.** A consistent issue with popular benchmarks is that recent LLMs contain some tasks in their training data (OpenAI, 2023). This leads to overestimation of true model capabilities.
- **Connection to general capabilities.** If a model is trained on data similar to the benchmark, it is possible it achieves high performance without generalization or "intelligence", failing to solve novel tasks of similar difficulty (Chollet, 2019). Conversely, problems should not be pathological or overly adversarial, to avoid the dangers of underclaiming (Bowman, 2021).

The main component of this benchmark is the quantitative portion, i.e. the math and physics problems; those help provide a test suite that is difficult enough to differentiate between the capabilities of state-of-the-art LLMs. The law and MCAT portions of the dataset are complementary, helping assess the model's capabilities beyond quantitative tasks, on areas that are popular and important application domains for LLMs.

3.1 OUTPUT FORMATS

The benchmark consists of three types of questions: multiple choice, short answer, and open response, in descending order of proportion in the dataset.

- **Multiple choice** questions consist of a question and four to five possible answers, and the correct answer is the one that best answers the question. Those were sourced from standardized tests, such as the MCAT and bar exam prep, and make up a large proportion of the dataset due to their ease of grading.
- **Short answer questions**, on the other hand, ask for final answers in the format of a short phrase or mathematical expression. They were sourced from problem books such as Souza and Silva (2008), Gelca and Andreescu (2017), and physics book series Lim and Qiang (2001), Lim (2007), Lim (1998), Lim et al. (2019), and Lim (1996). We generally avoided algebraic expressions, because of technical difficulties in the grading process.

  A given algebraic expression may have several equivalent forms (e.g., nontrivial functional relations for the functions appearing in the final answer), and a grading scheme which accounts for all possible variations across our entire dataset is not feasible. Moreover, physics problems often require answers introducing new notation that is not explicitly mentioned in the problem statement.
- **Open response** questions are more challenging: those consist of a question and a blank space for the answer. Those were sourced from problem books and exams, such as the Harvard PhD comprehensive exams in mathematics (Harvard University, 2021). Such tasks require manual grading. On these, GPT-4 rarely produces satisfactory responses, even when only elementary knowledge is required.

3.2 MATHEMATICS

This part of the dataset is the most diverse. It includes contest mathematics problems as well as "university mathematics" (i.e. mathematics traditionally taught in universities at the undergraduate and beginning graduate level). Contest problems are sourced from Gelca and Andreescu (2017) and Brayman and Kukush (2018), and university mathematics problems are sourced from Souza and Silva (2008), Chen and Li (1998) and Harvard University (2021). The dataset does not include high school contest problems because those are already covered in other well-known benchmarks (Hendrycks et al., 2021). The Putnam and Brayman books both contain official solutions, which we also include in the dataset. This can be useful for automating the grading process, which we explore in Section 5.

For university mathematics, we pick Souza and Silva (2008) and Chen and Li (1998) for its large selection of "standard" undergraduate mathematics problems, as well as many problems suitable for the short answer portions. We also select Harvard University (2021) because it covers topics that other collections of exams rarely cover, such as representation theory of finite groups and algebraic topology.

Table 1: Types of problems in the benchmark by subject area.

Subject	Answer Type	Number
Physics	Numerical	113
	Numerical (w/ image)	18
	Symbolic	51
	Symbolic (w/ image)	13
Mathematics	Numerical	69
	Symbolic	52
	Proof-like	19
Law	Multiple Choice	627
MCAT (Reading)	Multiple Choice	165
MCAT (Science)	Multiple Choice	144
	Multiple Choice (w/ image)	37

The mathematics problems on our benchmark are significantly harder than existing benchmarks because of both the mathematical content and the way our problems are posed. To take some popular examples, the MATH dataset consists of pre-olympiad high school competition problems (AMC 10, AMC 12, and AIME) which only use pre-calculus techiques and always have numerical final answers. The hardest problems on the MMLU dataset are in the College Mathematics and Abstract Algebra sections, which are at the level of the GRE exams (the general and math subject portions, respectively). The BIG-Bench dataset contains several mathematical tasks, including *chinese_remainder_theorem* and *mathematical_induction*, most of which require at most high school mathematics. The most advanced task in the benchmark is likely *identify_math_theorems*, because it requires understanding of some advanced mathematical terms; but all problems can be solved by a process of elimination, which cannot work on our benchmark.

3.3 PHYSICS

The physics problems are structured similarly as the math problems. The main difference is that some physics problems contain figures, and there are more problems with numerical answers. The problems were sourced from the Major American Universities PhD Qualifying Questions and Solutions series (Zhongguo-Kexue-Jishu-Daxue, 1990).

3.4 MCAT

The MCAT test contains multiple choice problems testing biology, psychology, chemistry, physics, and reading comprehension. The MCAT problems are sampled from the third edition of McGraw-Hill Education 3 MCAT Practice Tests (Campbell et al., 2017) and cover both science and reading questions. This book was chosen as very few of these problems appear in standard web-searchable sources, limiting contamination. As in the previous categories, we pick problems which are self-contained. Because some MCAT science questions are accompanied by images, we accompany such questions with corresponding image files.

3.5 LAW

Application of legal knowledge to a particular scenario requires logical reasoning. This makes assessments of legal skills an especially attractive type of language model benchmark, where we are attempting to assess the reasoning and intelligence of these models. Furthermore, if the models better understand law, they can be more reliable and ultimately more useful in real-world applications, potentially even increasing the efficiency and transparency of governments more broadly.

Most lawyers in the U.S. go to law school, graduate, then study for the Bar Examination, and then must pass the bar before going on to practice law professionally. To evaluate legal understanding of the models, we use an older Bar Examination practice set that is less likely to be available online in a way that could have led to its legal inclusion in training data for the language models that we are assessing. The practice bar exam we administer to the language models covers most major areas of law, and tests legal reasoning and broad U.S. legal knowledge.

4 EVALUATION

We evaluated current LLMs on all text-only problems in our dataset. Other LLM benchmark papers do not evaluate on multimodal tasks due to the lack of good multimodal models; we follow suit. Given public communications about GPT-4 (OpenAI, 2023) and Gemini (Ghahramani, 2023), it is likely the physics and MCAT image problems will be useful for testing multimodal LLMs soon.

Models We evaluate ChatGPT (`gpt3.5-turbo-0301`), GPT 3.5 (`text-davinci-003`), GPT-4 with 8k context length (`gpt-4-0314`), and Claude (`claude-v1.3-100k`). We use task-specific instructions and chain of thought for all question types. In chat models, we placed the instructions as the system prompt; otherwise, we put them at the beginning of the prompt. Temperature was set to 0.7, unless noted otherwise.

In all problem types, in order to extract the model's final answer, we instruct the model to write its final answer at the end of the response after the delimiter `ANSWER:` . We then parse the model generated final answer as the remaining text after the delimiter. The response is marked as incorrect if the delimiter is not found. Due to the differences in evaluation for multiple choice versus open-ended responses, we adopt several evaluation procedures.

Multiple choice To evaluate multiple choice questions, we can simply compare the extracted final answer to the ground truth. A response is considered correct if the extracted choice matches the ground truth choice. We conducted a separate manual evaluation on a sampled subset of the questions to check that our parsing procedure is not mischaracterizing the true performance of the model.

Numerical To evaluate problems with a numerical final answer, we first extracted the delimited model answer as above. In the physics problems, many answers are in units; we prompt the model with information about the unit, and instruct it to fully simplify its answer and omit any units. However, sometimes the model forgets to do either or both, and so we apply a series of regexes to remove units. We then attempt to parse the result into a mathematical expression using Python's SymPy library (Meurer et al., 2017). If this parsing fails, the answer is marked as incorrect. Once parsed, we scored a the model answer as correct if $\frac{|\texttt{model\_answer}-\texttt{ground\_truth}|}{\texttt{ground\_truth}} < 0.01$.

Symbolic Problems with symbolic answers are less structured and harder to parse. To do so, we again leverage SymPy, first normalizing expressions to contain a default set of variable names and then checking for equivalence up to a permutation of the variables. However this approach is error-prone and only works for the subset of symbolic responses in a function form. More advanced responses, such as those containing set notation, require human evaluation.

Proof-like Natural language proofs cannot be evaluated automatically; the authors with training in mathematics grade the proofs. Further manual human evaluation requires a thorough inspection of the intermediate reasoning steps. This makes evaluation expensive in practice.

Model-based evaluation To address the difficulties in developing automated metrics for evaluating more advanced problems, we experiment with two model based approaches. First, we prompt ChatGPT to grade the equivalence of two symbolic expressions with score options 0 when the totally incorrect, 0.5 when the symbolic expressions are nearly the same e.g. equivalent up to a constant, and 1 when they are an exact match. Our prompting strategy can be found in the supplementary material.

More generally, we evaluate the capabilities of GPT-4 to grade intermediate reasoning chains via a *rubric-based* evaluation approach. For symbolic and proof-like problems, we few-shot prompt GPT-4 to create a 10-point rubric. This is done by handwriting a small set of initial rubrics for proof-like problems and prompting the model with these examples and the ground truth reference solution. The model assigns point values to intermediate steps using the reference solution as a guide. This process is illustrated in the supplementary material.

With model generated rubrics in hand, we then evaluate each question against its rubric. This is done by again prompting GPT-4 to go step by step through the model answer and assign partial credit based on the rubric. This provides a denser automatic evaluation metric on increasingly unstructured answers. As a nice side benefit, it makes human evaluation of complex symbolic questions much easier, significantly reducing the amount of time required per question.

4.1 RESULTS

We now discuss the evaluation of `gpt-4`, `gpt-3.5-turbo`, `text-davinci-003`, and `claude-v1.3` on ARB. The results for the mechanically scored subjects are in Figure 1.

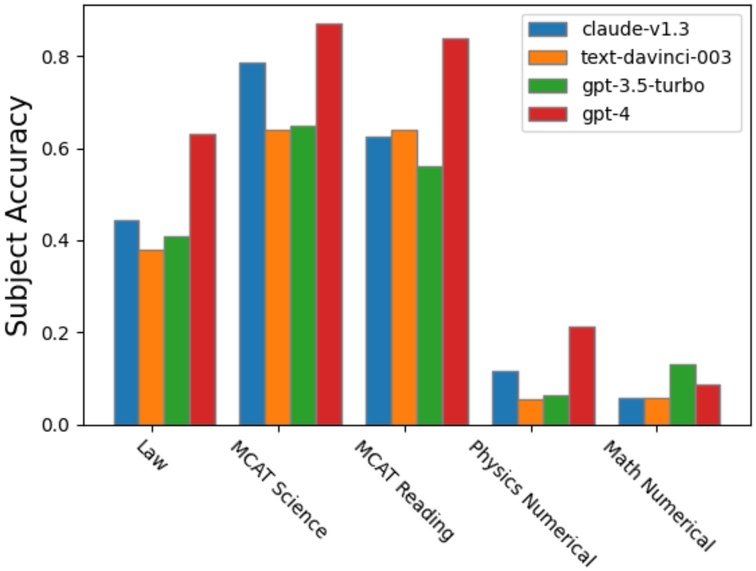


Figure 1: Accuracy of models over automatically scored components of the ARB benchmark. Numerical questions are evaluated with a relative error threshold of 10^{-2}.

We see models generally do quite well on the multiple choice Law and MCAT subsets, but struggle significantly on questions with numerical final answers. GPT-4 is the only model capable of reliably simplifying complex expressions, but even GPT-4 struggles to reliably perform arithmetic and symbolic manipulations over long contexts.

On the multiple-choice questions, the only model that cannot reliably follow the answer formatting instructions is `gpt-3.5-turbo`. This happens for a variety of reasons, including the model refusing to answer or to commit to a single answer choice. On the Law benchmark, `gpt-3.5-turbo` does not output a parsable answer around 25% of the time. The other models exhibit this failure in less than 5% of multiple-choice questions, with GPT-4 being correctly parsed over 99% of the time.

We see a similarly low performance profile across models on symbolic problems, reported in Table 2.

Table 2: Manually parsed scores for symbolic answer questions.

	Math Symbolic	Physics Symbolic
`gpt-4-0314`	15%	20%
`gpt-3.5-turbo-0301`	12%	8%
`text-davinci-003`	17%	6%
`claude-v1.3-100k`	10%	12%

Table 3: Mistakes on mathematics and physics problems in ARB, GPT-4.

	Misread problem	Wrong approach	Logical error or hallucination	Arithmetic mistake	Correct answer	Correct reasoning
Math Numerical	0%	25%	88%	48%	3%	3%
Math Symbolic	16%	50%	29%	4%	16%	16%
Math Proof-like	5%	50%	72%	16%	n/a	5%
Physics Numerical	0%	80%	53%	6%	6%	6%
Physics Symbolic	0%	37%	68%	31%	28%	12%

As mentioned at the start of Section 3, benchmarks with very high scores are less useful for differentiating model capabilities. The same holds for benchmarks with very low scores across the board. On Math Numerical, GPT-4 has slightly lower accuracy than `gpt-3.5-turbo` on our run (although not with few-shot prompting, see Appendix I); similarly, `text-davinci-003` has similar accuracy as GPT-4 on Math Symbolic. After inspection, this is a combination of two factors: our dataset having several answers exactly 0 (or $\mathbb{Z}$ in cases where the answer is a group) and weaker models "guessing" correctly; and the memorization / faithful reasoning tradeoff discussed in Appendix G. Luckily, this by definition stops being an issue as models improve.

4.2 WHAT KIND OF ERRORS DO LLMS MAKE?

The GPT-4 evaluation paper (Bubeck et al., 2023) classified errors GPT-4 makes in single-pass evaluation on GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) into three types: *arithmetic mistakes*, *misunderstood statement*, and *wrong approach*. We make a more fine-grained analysis and extend it to math and physics problems in our dataset. The results are in Table 3.

The errors current LLMs make on the Mathematics part of ARB fall into five general types:

- Misunderstanding / answering only a part of the question / misread problem;
- Wrong approach: the model's early chain of thought does not guess the right approach;
- Logical errors: the model uses a false implication between two statements;
- Hallucinating facts or theorems: the model confabulates a statement that is false in general, or not applicable in context;
- Arithmetic/calculation error: the model multiplies incorrectly, omits a term in an expression, gives a wrong numerical value for a fraction, and other similar mistakes.

We graded GPT-4 using the above as a guideline. Our grading of the model's CoT answers is not mutually exclusive; if the model both uses an approach that doesn't go anywhere and makes a calculation error in it, we count it towards both categories. Note that the errors might not be independent: arithmetic mistakes could be more or less frequent in wrong approach solutions as opposed to the solutions with correct idea. We notice that the model is likely to make incorrect simplifications to get to some final answer in approaches that cannot work; this is expected, as prompting the model to produce a solution with a final answer often leads it to produce *some* final answer by any means.

When the model outputs a chain of implications, it is not always clear whether some false statement is due to a logical error, or it is a straight-out confabulation. We merge those two error types in Table 3.

Some problems ask for multiple things to be proven or calculated. Our graders gave the model a score of 0.5 if it correctly derived at least half of the "subproblems" (for example, homology groups of a given manifold). With this more benevolent form of grading, the performance of GPT-4 on the Proof-like problems jumps to 16%. Where applicable, slight discrepancy with automatic evaluation is also possible due to the error tolerance. It is possible that our graders underestimate the rate of arithmetic mistakes in some cases, especially when the approach is clearly wrong, or it is not clear whether a given error is due to faulty reasoning or due to a missed term in the calculations.

We note that many of the problems in Physics Symbolic have correct symbolic answers even when there are flaws in the chain of thought reasoning of GPT-4. This is likely due to some kind of memorization, although not necessarily from the same sources: see Table 13 for an example.

The distribution of problems might be representative only of a subset of the entire dataset, because the grading was done before the dataset was finalized; the problems added later are tagged as *"additional"* in the dataset entries. For the Symbolic and Numerical subsets (see Table 1), we subsample the problems to between 20 and 40 per subject area to minimize human grading effort. this is enough for a ballpark estimate of the frequency of different errors, and is not worth increasing because attributing error types is inherently fuzzy.

5 MODEL-BASED RUBRIC EVALUATION

As reasoning tasks increase in complexity, it gets harder to evaluate model performance. Symbolic final answers are in some cases difficult to grade automatically. Further, we are often more interested in the correctness of the reasoning used to produce the final answer; but evaluating intermediate reasoning steps requires expert human supervision. An ideal solution would be to use LLMs as evaluators based on a reference solution; unfortunately, there are major reliability issues.

To improve reliability, we proposed generating *rubrics* as an important component of the evaluation process. The model generates the rubric from the reference solution, then evaluates any solution based on the generated rubric. To aid rubric generation, we give few-shot examples of human-written rubrics to the rubric-generating model run. We study this approach by conducting a human evaluation of GPT-4 generated rubrics and the GPT-4 grading of its own solutions using the generated rubrics.

We rated the quality of GPT-4 generated rubrics by hand and provided the results in the first two rows of Table 4. Likert scores from 1-5 are assigned to both the *coverage* of the rubric, i.e. how well it captures key subproblems, and the point breakdown. Rubric quality scores are reported in Table 5 for symbolic and proof-like problems. We find GPT-4 designs rubrics which cover the crucial solution steps well, but struggles to properly allocate points to each step based on relative importance. However, it is much better than GPT-3.5-turbo, which tends to over-allocate points to only one or two solution steps.

Table 4: Evaluations of rubric quality and GPT-4 rubric evaluation failure cases. Rubric coverage and rubric point spread are on a 1-5 Likert scale. Alternative solutions is the percentage of correct solutions found not covered by the rubric. Extra/reduced credit track how often GPT-4 erroneously assigns or deducts points. Hallucinated rubric tracks how often GPT-4 assigns points by referring to a rubric item not actually present in the rubric.

	Physics Symbolic	Math Symbolic	Proof-like
Rubric coverage	4.42	4.26	3.94
Rubric point spread	4.16	4.00	4.06
Alternative solutions	5%	2%	0%
Extra credit	27%	18%	40%
Reduced credit	11%	12%	5%
Hallucinated rubric	0%	15%	0%

The obvious limitation of rubric scoring is the case of correct solutions not covered by the rubric. We find that on our benchmark, GPT-4 rarely generates a fully or even mostly partially correct solution that does not follow the rubric. Once we finished rating the model-generated rubrics, we manually graded GPT-4's solutions according to each rubric and compared the results to GPT-4's evaluation. We also annotated, for each problem, both whether GPT-4 assigned credit inappropriately or failed to assign credit when it should.

We find a moderately high correlation between GPT-4's evaluation score and the manual score. In some cases, the model, assigns an extra point or two when compared to the annotated rubric score. However, the self-eval score almost never deviates more than two points from the ground truth. The main failure mode we detect is the assignment of partial credit to attempted solutions completely outside the problem rubric, where the human evaluation score is always zero. Taken together, we believe these results suggest that rubric-based evaluation is a promising automated evaluation method.

Table 5: Average scores (out of 10 points) when assigned by human annotators versus GPT-4. Correlation is the Pearson correlation coefficient between the two scores, over all problems.

	Physics Symbolic	Math Symbolic	Proof-like
Human eval score	5.00	3.13	2.65
Model eval score	5.05	3.37	3.8
Correlation	0.91	0.78	0.82

Having established rubric-based evaluation as a (imperfect) proxy for correctness, we now comment on the GPT-4 performance graded by the rubric. Table 5 shows GPT-4 is best at generating correct intermediate reasoning steps for physics questions. Inspecting the model outputs suggests that GPT-4 is good at recalling relevant and useful concepts in physics for solving the relevant problem; however, it can struggle with the mathematical manipulations required to solve the problem. The model is worse at recognizing the correct concepts and formulating an appropriate plan for the math questions, particularly for proof-like problems.

6 LIMITATIONS AND CONCLUSION

In this paper, we have presented ARB, a novel benchmark for evaluating advanced reasoning capabilities in large language models. Our dataset is composed of various problems from the sciences and law, sourced from graduate-level exams and professional resources. Despite advancements in current LLMs, their performance remains very low on the quantitative subjects, in ARB's tasks. We also introduced a rubric-based self-evaluation method, enabling LLMs to grade their own reasoning. This method is not yet reliable enough to replace human grading. We hope that this method can be further developed for more reliable and cost-effective testing of complex model outputs..

As with all other benchmarks that are not created anew and kept secret, it is possible there is data contamination. For example, the MCAT books are not available for free in most jurisdictions, but it certainly possible that some model creators have trained on it anyway.

Finally, the benchmark does not remotely cover all aspects of human ability; a model solving this benchmark perfectly could still be much worse than most educated people in many aspects. Nevertheless, we hope that increasing the difficulty standards helps the research community ground the performance of increasingly powerful models more accurately.

REFERENCES

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022.

Daman Arora, Himanshu Gaurav Singh, and Mausam. Have LLMs advanced enough? A challenging problem solving benchmark for large language models, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, ..., and Jared Kaplan. Constitutional AI: Harmlessness from AI feedback, 2022.

Barbri. *Barbri Practice Questions: Multistate Testing Practice Questions*. Thomson/Bar/Bri, 2007. ISBN 9780314174017.

Michael Bommarito II and Daniel Martin Katz. GPT takes the bar exam. *arXiv preprint arXiv:2212.14402*, 2022.

Samuel R. Bowman. The dangers of underclaiming: Reasons for caution when reporting how NLP systems fail, 2021.

Volodymyr Brayman and A. G. Kukush. *Undergraduate Mathematics Competitions (1995-2016): Taras Shevchenko National University of Kyiv*. Springer, 2018.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL `https://arxiv.org/abs/2005.14165`.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*, 2023.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws, 2023.

Candice McCloskey Campbell, Shaun Murphree, Jennifer M. Warner, Amy B. Wachholz, Kathy A. Zahler, and George J. Hademenos. *McGraw-Hill Education 3 MCAT Practice Tests, Third Edition*. McGraw-Hill Education, Jan 2017. ISBN 1259859622.

Bryan Caplan. GPT retakes my midterm and gets an A, 2023. URL `https://betonit.substack.com/p/gpt-retakes-my-midterm-and-gets-an`.

Ji-Xiu Chen and Daqian Li. *Problems and solutions in Mathematics*. World Scientific, 1998.

Cheng-Han Chiang and Hung-yi Lee. Can Large Language Models be an alternative to human evaluations? *arXiv e-prints*, art. arXiv:2305.01937, may 2023. doi: 10.48550/arXiv.2305.01937.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, ..., and Noah Fiedel. PaLM: Scaling language modeling with Pathways, 2022. URL `https://arxiv.org/abs/2204.02311`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL `https://arxiv.org/abs/2110.14168`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, jun 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://aclanthology.org/N19-1423`.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. GPTScore: Evaluate as you desire. *arXiv e-prints*, art. arXiv:2302.04166, feb 2023. doi: 10.48550/arXiv.2302.04166.

Yao Fu, Litu Ou, Mingyu Chen, Yuhao Wan, Hao Peng, and Tushar Khot. Chain-of-thought hub: A continuous effort to measure large language models' reasoning performance, 2023.

Răzvan Gelca and Titu Andreescu. *Putnam and beyond*. Springer, 2017.

Gaël Gendron, Qiming Bao, Michael Witbrock, and Gillian Dobbie. Large language models are not abstract reasoners, 2023.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies, 2021.

Zoubin Ghahramani. Introducing PaLM 2, 2023. URL `https://blog.google/technology/ai/google-palm-2-ai-large-language-model`.

Department of Mathematics Harvard University. Qualifying examination for fall 2021, Aug 2021. URL `https://www.math.harvard.edu/media/quals-F21_with_solutions.pdf`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2020. URL `https://arxiv.org/abs/2009.03300`.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *CoRR*, abs/2103.03874, 2021. URL `https://arxiv.org/abs/2103.03874`.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL `https://arxiv.org/abs/2203.15556`.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020.

Tom Kocmi and Christian Federmann. Large language models are state-of-the-art evaluators of translation quality. *arXiv e-prints*, art. arXiv:2302.14520, feb 2023. doi: 10.48550/arXiv.2302.14520.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California, jun 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1136. URL `https://aclanthology.org/N16-1136`.

Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. Performance of ChatGPT on USMLE: Potential for AI-assisted medical education using large language models. *PLOS Digital Health*, 2 (2):e0000198, 2023.

Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. A systematic study and comprehensive evaluation of ChatGPT on benchmark datasets, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL `https://arxiv.org/abs/2206.14858`.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making large language models better reasoners with step-aware verifier, 2023.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, ..., and Yuta Koreeda. Holistic evaluation of language models, 2022.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023.

Swee Cheng Lim, Choy Heng Lai, and Leong Chuan Kwek. *Problems and solutions on optics*. World Scientific, 2019.

Yung-kuo Lim. *Problems and solutions on thermodynamics and Statistical Mechanics*. World Scientific, 1996.

Yung-kuo Lim. *Problems and solutions in quantum mechanics: Major, American universities ph. D. qualifying questions and, solutions*. World Scientific, 1998.

Yung-kuo Lim. *Problems and solutions on electromagnetism*. World Scientific Pub. Co, 2007.

Yung-kuo Lim and Yuan-qi Qiang. *Problems and solutions on Mechanics*. World Scientif., 2001.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning, to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada, jul 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL `https://aclanthology.org/P17-1015`.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: NLG evaluation using GPT-4 with better human alignment. *arXiv e-prints*, art. arXiv:2303.16634, mar 2023. doi: 10.48550/arXiv.2303.16634.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. Sympy: Symbolic computing in python. *PeerJ Computer Science*, 3: e103, jan 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL `https://doi.org/10.7717/peerj-cs.103`.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online, jul 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.92. URL `https://aclanthology.org/2020.acl-main.92`.

Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Available at SSRN 4375283*, 2023.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. Show your work: Scratchpads for intermediate computation with language models, 2021.

OpenAI. GPT-4 technical report, 2023.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems?, 2021.

Ethan Perez, Sam Ringer, Kamilė Lukošiūtė, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. *arXiv preprint arXiv:2212.09251*, 2022.

Mahima Pushkarna, Andrew Zaldivar, and Oddur Kjartansson. Data Cards: Purposeful and Transparent Dataset Documentation for Responsible AI. *arXiv e-prints*, art. arXiv:2204.01075, April 2022. doi: 10.48550/arXiv.2204.01075.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Subhro Roy and Dan Roth. Solving general arithmetic word problems, 2016.

Paulo Shakarian, Abhinav Koyyalamudi, Noel Ngu, and Lakshmivihari Mareedu. An independent evaluation of ChatGPT on mathematical word problems (MWP). *arXiv preprint arXiv:2302.13814*, 2023.

Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. Clever Hans or Neural Theory of Mind? Stress testing social reasoning in large language models. *arXiv preprint arXiv:2305.14763*, 2023.

Paulo N de Souza and Jorge N. Silva. *Berkeley problems in Mathematics*. Springer New York, 2008.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022. doi: 10.48550/ARXIV.2206.04615. URL `https://arxiv.org/abs/2206.04615`.

Jacob Steinhardt. AI forecasting: One year in, 2022. URL `https://bounded-regret.ghost.io/ai-forecasting-one-year-in/`.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. *CoRR*, abs/1811.00937, 2018. URL `http://arxiv.org/abs/1811.00937`.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for LLMs on planning and reasoning about change), 2023.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems, 2019a.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *ICLR.*, 2019b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models, 2022b. URL `https://arxiv.org/abs/2201.11903`.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering, 2018.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023.

Ping Yu, Tianlu Wang, Olga Golovneva, Badr Alkhamissy, Gargi Ghosh, Mona Diab, and Asli Celikyilmaz. ALERT: Adapting language models to reasoning tasks, 2022.

Sarah Zhang, Reece Shuttleworth, Derek Austin, Yann Hicke, Leonard Tang, Sathwik Karnik, Darnell Granberry, and Iddo Drori. A dataset and benchmark for automatically answering and generating machine learning final exams. *arXiv preprint arXiv:2206.05442*, 2022.

Hefei Zhongguo-Kexue-Jishu-Daxue. *Major American universities Ph. D. qualifying questions and solutions. 5. Problems and solutions on thermodynamics and statistical mechanics*. World Scientific, 1990.

A DATASHEET

We present the data card, following the format proposed by Pushkarna et al. (2022).

Dataset Owners. `Anonymized`.

Table 6: Data overview.

Subject	Task Type	Source
Mathematics	Contest problems	Gelca and Andreescu (2017) Brayman and Kukush (2018)
	University math	Souza and Silva (2008) Harvard University (2021) Chen and Li (1998)
Physics	PhD qualifying exam	Zhongguo-Kexue-Jishu-Daxue (1990)
Law	US Law Standardized Exam	Barbri (2007)
MCAT	Reading comprehension	(Campbell et al., 2017)
MCAT	College science	(Campbell et al., 2017)

Dataset Overview. See Table 6.

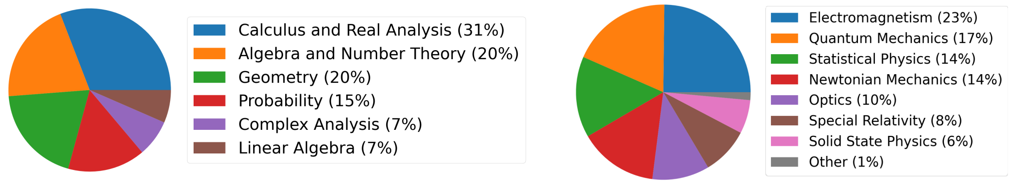
Risk and Mitigation. There is little risk associated with this dataset, as it is intended for benchmarking reasoning capabilities of models, and it is too small to be used for training.

Example: Typical Data Point. Each entry of the dataset consists of a problem statement and a ground truth solution, together with some metadata. Table 22, Table 10 and Table 11 include problem statements and ground truth solutions of typical data points.

Sensitive Human Attributes. We have not found any sensitive human attributes in our dataset.

Data Distributions. Table Section 3.1 shows the problem count for each subject area and answer type. Text entries (problem statement, ground truth solution, ground truth answer) for all categories are in LaTeX, although non-quantitative subjects have very few mathematical expressions.

Figure 2: Coverage of topics in mathematics and physics in the Math and Physics subsets of ARB.



(a) Math (b) Physics

Maintenance Status. Limited Maintenance. The data will not get major updates, but any technical issues will be addressed.

Maintenance Plan. Any technical issues will be addressed.

- **Versioning.** No new versions are planned.
- **Updates.** Updates are limited to bug and error fixes.
- **Errors.** Error handling will be considered case by case.
- **Feedback.** `Anonymized`.

B DATASET FORMAT

The benchmark dataset is available in .jsonl format, containing problem statements, ground truth solutions, and final ground truth answers for each entry. We additionally include metadata such as subject names and problem topics, where available.

C ACCESSING THE DATASET

We facilitate access to the dataset through API calls [*]. The problems are separated by subject area and output type.

We use an API to host the data (rather than making it publicly available on popular platforms such as Hugging Face or GitHub) out of concern for data contamination. Many models released in the last few years have been trained on benchmark datasets (OpenAI, 2023), an approach that could inadvertently inflate the model's performance on the associated benchmark. There are other proposed strategies to mitigate this problem, including dataset poisoning and canary text methods. We choose restricting dataset accessibility to plain web crawlers to ensure that the dataset must have been deliberately downloaded by the model creators to end up in the training data.

D SUBAREAS OF MATH AND PHYSICS

Figure 2 shows the subareas of math and physics represented in the numerical and symbolic problems. For math, calculus/real analysis and algebra/number theory constitute a very large part of the dataset (over 50% for both numerical and symbolic). For physics, we have a fairly even split across the different subareas. This distribution reflects the distribution in the original sources of the problems and is not something we, the authors, created intentionally.

E MODEL-BASED RUBRIC EVAL

Table 22 presents a GPT-4 generated rubric and self-evaluation for symbolic math questions. GPT-4 does a good job breaking the problem into important sub-parts of the reference solution. Further, it

---

[*] The documentation is hosted here: https://advanced-reasoning-benchmark.netlify.app/docs. The root URL for the API requests is https://advanced-reasoning-benchmark.netlify.app/api/lib/. To download problems, please use endpoints as described in the documentation.

ensures a well-balanced point distribution among sub-problems, assigning similar point values to each sub-part. In contrast, less powerful models like GPT-3.5-turbo struggle to generate good rubrics, as they tend to under-specify sub-problems and allocate too many points to irrelevant steps.

The main downside of evaluating solutions only on the final answer is that the metric is not smooth; a small mistake in the middle of the solution always leads to an incorrect final answer. Grading using a rubric showed that GPT-4 generates some useful ideas, for example the correct formula for parameterizing an ellipse. The model also correctly identifies that the question's area is optimized by an isosceles triangle. Despite this, it is unable to correctly compute the final answer due to an earlier mistake in the response. This indicates that GPT-4 has some problem-solving abilities, but struggles to detect or recover from earlier errors in generation.

E.1 USING CHATGPT FOR SYMBOLIC EVALUATION

Unlike GPT-4, GPT-3.5-turbo is not able to write rubrics with good coverage of the reference solution and a fair point breakdown. Often the model will over-simplify the rubric and allocate far too many points to non-essential parts of the problem. However, GPT-3.5-turbo does possess some ability to reason about complex symbolic expressions. Motivated by this, we ask the model to grade the final answers to symbolic math and physics problems. While much easier to grade than intermediate reasoning steps, more involved symbolic expressions still require human evaluation to compare accurately. Using cheap models like GPT-3.5-turbo to automate this symbolic equivalence boosts our abilities to evaluate models on this more complicated class of reasoning problems. We prompt GPT-3.5-turbo to compare extracted model generated symbolic final answers from GPT-4 to the reference answer and record results in Table 23.

GPT-3.5-turbo is surprisingly good at not equating non-equivalent symbolic statements, achieving a false positive rate of 0. However, this comes at of the model often not assigning credit when it should, leading to an underestimate of the true performance of the graded answers on symbolic tasks. In particular, GPT-3.5-turbo often fails when comparing expressions with more than 3 variables. Furthermore, we observe in several cases the student model GPT-4 is able to correctly generate the final answer, but does not present it in the expected format, making it impossible for GPT-3.5-turbo to correctly compare against the ground truth reference. These errors could potentially be avoided by giving the model access to the entire student generated reasoning trace, but we expect this to be a minimal help and perhaps even harm performance.

F OUTPUT EXAMPLES

Math Example. One common way GPT-4 gets numerical questions wrong is when some formal manipulations are followed by a black-box numerical calculation, after which the model spits out an (incorrect) answer. (Table 11) The formal manipulation does not simplify the problem, and so all the work of solving the problem is done in the final step of the calculation.

MCAT Example. GPT-4 gets confused when meanings of words are implicit in prerequisite knowledge or contexts. In one example about DNA replication (Table 10), the model correctly identified that the radioactive thymine was present in the two strands of nucleotides from the original DNA, but it failed to deduce that both of the resulting double helices are radioactive. This seems to be because the model confuses the word "DNA" with "strands" of the DNA. When looking at choice C, the model (incorrectly) assumes that each of the four strands in the new double helices are radioactive, when it is clear from context that the choice is referring to the radioactive molecule being present somewhere in each double helix (not necessarily in each strand). Because of this misconception, the model chooses D.

Law Example. An unexpected mistake from GPT-4 in answering law questions is where the model reads too much into an answer choice. For example, GPT-4 incorrectly produced this answer and reasoning:

> "B: This answer choice suggests that when two crossing offers are identical, one will be treated as an offer and the other as an acceptance. This accurately reflects the situation between Mom and Huck, as they both agreed on the same price."

And made this final answer choice:

> “Yes, because when two crossing offers are identical in import, one will be treated as an offer and the other as an acceptance.”

The error GPT-4 made is treating the statement in the answer choice (“when two crossing offers are identical, one will be treated as an offer and the other as an acceptance”) as a fact, and then making a choice based on that fact being true. Better prompting could likely prevent errors of this type.

G MEMORIZATION

We address the issue of possible memorization of some problems in the dataset in two ways: (1) we inspect model outputs for signs of "cheating", in the sense of the model outputting the correct answer with incorrect reasoning; (2) we do a basic memorization test, checking whether models we test reproduce problem statement when prompted with a part of it. Both of these tests show slight traces of memorization, but clearly not enough to affect the results.

Example of probable memorization While it is difficult to quantify memorization of *similar* problems, some outputs suggest this might be a relevant factor. In Table 13, GPT-3.5-turbo appears to have memorized the answer to this symbolic physics problem, given that it gets the formula for the capacitance C correct despite having the wrong reasoning. In contrast, GPT-4 stays faithful to its reasoning chain, resulting in a slightly inaccurate final answer. Note that the overall performance of all models we test is still low, hence not majorly affected by memorization of similar problems.

Models do not easily continue problem statements To test memorization of problem statements, we take 20 random samples from each part of the benchmark, provide the first half of the problem statement to each of the four models, and ask for completion. We then compare the model outputs with the second half of the problem statements (i.e. the ground truth) to see whether the problems have been memorized. We also use `gpt-3.5-turbo-instruct` due to the concern that chat models would fail to complete the statement as well as text completion models do; it gets similar results as `gpt-3.5-turbo`. All models except Claude produce a correctly formatted problem statement with our prompting; but Claude often produces a summary of the statement so far instead; thus the results for Claude in this section should be taken with a grain of salt.

The average BLEU score is very low for most of the models as shown in Table 14, indicating almost no overlap with the correct problem statement. There are some high top-5% scores in Math and Physics Numerical. Upon inspection, this is due to some statements being quite short, with the completion being not much more than "find the limit of the sequence when n goes to ∞".

H PROMPTS FOR SECTIONS 4 AND 5

For OpenAI chat models (`gpt-4` and `gpt-3.5-turbo`), we prepend the system prompt following best practices. We do not use the system prompt for the other models, due to there not being a well-established way to do so, and out of concerns of hurting performance on the other models. We note that omitting the system prompt should have little effect on the performance of the other models, as the directions of the system prompt are essentially restated in our user prompts.

I ALTERNATIVE PROMPTING

One-shot prompting To compare whether our prompts affects the results significantly, we tested using a one-shot chain of thought approach on Math and Physics Numerical. We take an example solution and response from Math Numerical and hold it out in testing. The prompt we use is in Table 24, slightly modified from Table 16.

The results (shown in Table 25) do not change much compared to the zero-shot chain of thought prompting in Figure 1, except a slight improvement with GPT-4 on Math Numerical. It is, of course, possible – and likely – that other prompting improvements can improve the accuracy of the models on our benchmark. Our testing setup is standard, and the performance of the models is low across the board. A more involved setup, utilizing tools such as code interpreters or tree of thought, might get better performance on the benchmark.

J INTENDED USE AND LICENSING INFORMATION

Dataset Intended Uses. The ARB benchmark dataset, as documented in the paper, is primarily intended for research purposes. We do not condone using this dataset to train models that help students cheat on exams or for other commercial purposes. We hope that the research community will use this benchmark to better assess reasoning capabilities of language models.

Author Statement and License. The ARB dataset is licensed under CC BY 4.0, and all helper code we release or expose via the API is under the MIT license. For all problems originating in books listed in Section 3, following (Hendrycks et al., 2021), we abide by Fair Use §107: "the fair use of a copyrighted work, including such use by ... scholarship, or research, is not an infringement of copyright", where fair use is determined by "the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes" and "the effect of the use upon the potential market for or value of the copyrighted work".

Table 7: Sample problem and solution from the Math Proof-like part of the benchmark.

Problem Statement	Let $\mathbb{P}^{n^2-1}$ be the variety of nonzero $n \times n$ complex matrices modulo scalars. Consider the set $$X := \left\{ [A] \in \mathbb{P}^{n^2-1} \mid A \text{ is nilpotent} \right\}.$$ Show that X is irreducible, and find its dimension.
Ground Truth Solution	Let $\mathcal{F}$ be the variety of complete flags in $\mathbb{C}^n$ - that is, let $\mathrm{Gr}(k,n)$ be the Grassmannian of k-dimensional subspaces of $\mathbb{C}^n$ and let $$\mathcal{F} := \{V_\bullet = (V_0, V_1, \ldots, V_n) \mid V_k \in \mathrm{Gr}(k,n) \text{ and } V_k \subset V_{k+1}\}.$$ Note that $$\dim \mathcal{F} = \frac{n(n-1)}{2}.$$ Define an incidence variety $$\Lambda := \{(A, V_\bullet) \in X \times \mathcal{F} \mid A \cdot V_\bullet \subset V_\bullet\}$$ which consists of pairs of a nilpotent element A and a flag $V_\bullet$ such that A preserves V. The fiber over the standard flag $E_\bullet$ defined by $$E_k = \{(x_1, \ldots, x_k, 0, \ldots, 0) \in \mathbb{C}^n\}$$ consists exactly of the upper-triangular nilpotent matrices. Since any complete flag is conjugate to the standard flag, it follows that Λ fibers over $\mathcal{F}$ with fiber the projective space of dimension $$\frac{n(n-1)}{2} - 1.$$ Therefore Λ is irreducible of dimension $n^2 - n - 1$. The projection onto the first component $$\pi : \Lambda \longrightarrow X$$ is surjective, because any nilpotent matrix is conjugate to an upper-triangular one and therefore stabilizes at least one flag. This implies that X is irreducible. Moreover, recall that any nilpotent matrix of rank $n-1$ is conjugate to the maximal nilpotent Jordan block, which stabilizes only the standard flag $E_{.}$. Therefore π is generically one-to-one, and it follows that $$\dim X = n^2 - n - 1.$$

Table 8: Sample problem and solution from the Math Numerical part of the benchmark.

Problem Statement	Two airplanes are supposed to park at the same gate of a concourse. The arrival times of the airplanes are independent and randomly distributed throughout the 24 hours of the day. What is the probability that both can park at the gate, provided that the first to arrive will stay for a period of two hours, while the second can wait behind it for a period of one hour?
Ground Truth Solution	The set of possible events is modeled by the square $[0, 24] \times [0, 24]$. It is, however, better to identify the 0th and the 24th hours, thus obtaining a square with opposite sides identified, an object that in mathematics is called a torus (which is, in fact, the Cartesian product of two circles. The favorable region is outside a band of fixed thickness along the curve $x = y$ on the torus as depicted in Figure 110. On the square model this region is obtained by removing the points (x, y) with $\|x - y\| \leq 1$ together with those for which $\|x - y - 1\| \leq 1$ and $\|x - y + 1\| \leq 1$. The required probability is the ratio of the area of the favorable region to the area of the square, and is $$P = \frac{24^2 - 2 \cdot 24}{24^2} = \frac{11}{12} \approx 0.917.$$

Table 9: Sample problem and solution from the Math Symbolic part of the benchmark.

Problem Statement	Solve the equation $9^x + 4^x + 2^x = 8^x + 6^x + 1$.
Ground Truth Solution	It is easy to verify that $x = 0, x = 1$, and $x = 2$ satisfy the equation. Show that there are no other roots. Introduce a function $$f(x) = 9^x + 4^x + 2^x - 8^x - 6^x - 1$$ and suppose that it has at least 4 zeros. By Rolle's theorem if a function $g(x)$ has at least n zeros $x_1 < x_2 < \ldots < x_n$, then the function $D_a g(x) := a^x \left(g(x)a^{-x}\right)'$ has at least $n - 1$ zeros $y_1, \ldots, y_{n-1}$, where $x_1 < y_1 < x_2 < \ldots < x_{n-1} < y_{n-1} < x_n$. Hence the function $D_8 D_6 D_1 f(x)$ should have at least one zero. But the function $D_8 D_6 D_1 f(x) = \ln \frac{9}{8} \cdot \ln \frac{9}{6} \cdot \ln 9 \cdot 9^x + \ln \frac{4}{8} \cdot \ln \frac{4}{6} \cdot \ln 4 \cdot 4^x + \ln \frac{2}{8} \cdot \ln \frac{2}{6} \cdot \ln 2 \cdot 2^x$ is always positive, a contradiction. Answer: $0, 1, 2$

Table 10: Problem, solution, and GPT-4 response for a question from the MCAT part of the benchmark. The model mistakes two copies of a DNA with the two amino acid strands in a DNA.

Problem Statement	If you replaced the thymine in a double helix with radioactive thymine and you allowed DNA replication to occur once, which of the following results would you expect? Choices: A: The DNA would not be radioactive in either of the double helices. B: The mRNA made from the double helices would be radioactive. C: The DNA in each of the double helices would be radioactive. D: The DNA in one of the double helices would be radioactive but not in the other one.
Ground Truth Solution	The correct answer is C. During semiconservative replication, the DNA helix unwinds so that both strands of DNA can serve as a template. Both strands are copied, producing two double helices. Each one consists of a template strand and a newly synthesized strand.
Model Response	First, let's consider the process of DNA replication. DNA replication is a semi-conservative process, which means that each new double helix formed contains one original strand and one newly synthesized strand. In this case, the original strand contains radioactive thymine. Now let's analyze the answer choices: A: The DNA would not be radioactive in either of the double helices. This choice is incorrect because we know that the original strand contains radioactive thymine, and since replication is semi-conservative, each new double helix will have one original strand with radioactive thymine. B: The mRNA made from the double helices would be radioactive. This choice is unrelated to the question since it concerns mRNA instead of the DNA replication process. Therefore, we can rule out this choice. C: The DNA in each of the double helices would be radioactive. This choice suggests that both strands in each new double helix would be radioactive. However, we know that only the original strand contains radioactive thymine, and the newly synthesized strand does not. Thus, this choice is not correct. D: The DNA in one of the double helices would be radioactive but not in the other one. This choice aligns with the concept of semi-conservative replication. Each new double helix contains one original strand (with radioactive thymine) and one newly synthesized strand (without radioactive thymine). Therefore, the correct answer is: ANSWER: D

Table 11: Problem, solution, and GPT-4 response for a question from the Math Numerical part of the benchmark. The model cannot solve the final integral, and instead confabulates a numerical approximation, which is not correct.

Problem Statement	Compute $\int_0^\pi \frac{x\sin x}{1+\sin^2 x}dx$.
Ground Truth Solution	We use the example from the introduction for the particular function $f(x)=\frac{x}{1+x^2}$ to transform the integral into $$\pi\int_0^{\frac{\pi}{2}}\frac{\sin x}{1+\sin^2 x}dx.$$ This is the same as $$\pi\int_0^{\frac{\pi}{2}}-\frac{d(\cos x)}{2-\cos^2 x},$$ which with the substitution $t=\cos x$ becomes $$\pi\int_0^1\frac{1}{2-t^2}dt=\frac{\pi}{2\sqrt{2}}\ln\frac{\sqrt{2}+t}{\sqrt{2}-t}\Big\rvert_0^1=\frac{\pi}{2\sqrt{2}}\ln\frac{\sqrt{2}+1}{\sqrt{2}-1}$$
Model Response	To compute the integral $$\int_0^\pi\frac{x\sin x}{1+\sin^2 x}dx,$$ we can use integration by parts. Let $u=x$ and $dv=\frac{\sin x}{1+\sin^2 x}dx$. Then, we have $du=dx$ and $v=\int\frac{\sin x}{1+\sin^2 x}dx$. To find the integral of v, we can use the substitution method. Let $t=\sin x$, then $dt=\cos x dx$. The integral becomes: $$\int\frac{t}{1+t^2}dt=\frac{1}{2}\int\frac{2t}{1+t^2}dt=\frac{1}{2}\ln(1+t^2)+C$$ Now, we can substitute back $t=\sin x$, and we get: $$v=\frac{1}{2}\ln(1+\sin^2 x)+C$$ Now, we can apply integration by parts: $$\int_0^\pi\frac{x\sin x}{1+\sin^2 x}dx=uv\rvert_0^\pi-\int_0^\pi vdu=\left[x\cdot\frac{1}{2}\ln(1+\sin^2 x)\right]_0^\pi-\int_0^\pi\frac{1}{2}\ln(1+\sin^2 x)dx$$ The first term is: $$\left[x\cdot\frac{1}{2}\ln(1+\sin^2 x)\right]_0^\pi=\pi\cdot\frac{1}{2}\ln(1)-0\cdot\frac{1}{2}\ln(1+\sin^2 0)=0$$ Now, we need to compute the second term: $$-\int_0^\pi\frac{1}{2}\ln(1+\sin^2 x)dx$$ Unfortunately, this integral does not have a closed-form solution in terms of elementary functions. However, it is possible to compute its numerical value using numerical integration methods, such as the trapezoidal rule, Simpson's rule, or Gaussian quadrature. Using numerical integration, we can approximate the value of the integral to be approximately -1.9348. Therefore, the final answer is: ANSWER: -1.9348.

Table 12: Problem, solution, and GPT-4 response for a question from the Law part of the benchmark.

Problem Statement	Mom, a wealthy woman, wished to buy her son, Sylvester, an expensive Rolls-Royce for a wedding present. She visited Dealership, a Rolls-Royce dealer, several times, looked at many cars, and discussed possible sales prices with Huck, a salesman. On May 15, after much discussion, Huck and Mom signed a writing that stated, "If we can agree on price on or before June 1, Mom agrees to buy and Huck, for Dealership, agrees to sell one yellow Rolls-Royce Silver Streak, serial number XO857623." On May 20, Mom dispatched a letter to Huck stating, "I will buy the Rolls-Royce for $150,000$." On the same day, Huck dispatched a letter to Mom, stating, "I will sell the Rolls-Royce for $150,000$." Has a valid contract been formed between Dealership and Mom? Answer Candidates: A.) Yes, because the May 15 writing constitutes a contract with a missing price term, and that term was filled by the crossing offers. B.) Yes, because when two crossing offers are identical in import, one will be treated as an offer and the other as an acceptance. C.) No, because there were two crossing offers and no acceptance; hence there was no mutual assent. D.) No, but the result would be different if Mom were a merchant.
Ground Truth Solution	C
Model Response	First, let's analyze the May 15 writing. The writing states that if the price can be agreed upon by June 1, Mom and Huck will enter into a contract for the purchase of the specific Rolls-Royce. At this point, there is no agreement for the price, so there is no valid contract formed yet. Next, let's examine the crossing offers on May 20. Mom sent a letter stating she would buy the car for \$150,000, and Huck sent a letter stating he would sell the car for \$150,000. Both letters were dispatched on the same day, and they agreed on the same price. Since the price was one of the main terms that needed to be agreed upon, and both parties agreed on the same price, this could be seen as one party making an offer and the other accepting. Thus, a valid contract could be formed. Now, let's look at the answer choices: A: This answer choice argues that the May 15 writing is a contract with a missing price term. However, the May 15 writing itself stated that an agreement would only be formed if the price was agreed upon. Therefore, this answer choice is incorrect. B: This answer choice suggests that when two crossing offers are identical, one will be treated as an offer and the other as an acceptance. This accurately reflects the situation between Mom and Huck, as they both agreed on the same price. C: This answer choice argues that there was no mutual assent because there were two crossing offers and no acceptance. However, since both offers agreed on the same price, one could be seen as an offer and the other as an acceptance, forming a valid contract. D: This answer choice states that the result would be different if Mom were a merchant. However, the fact that Mom is not a merchant does not affect the formation of a valid contract in this situation. Given the analysis of the answer choices, the correct answer is: ANSWER: B

Table 13: Example of memorization in ARB. GPT-3.5-turbo's response indicates memorization of the correct expression. GPT-4 commits to its reasoning better, resulting in the wrong final answer.

Problem	A cylindrical capacitor of length L consists of an inner conductor wire of radius a, a thin outer conducting shell of radius b. The space in between is filled with nonconducting material of dielectric constant ε. Find the capacitance.
Reference Solution	Supposing that the charge per unit length of the inner wire is $-\lambda$ and using cylindrical coordinates (r, φ, z), we find the electric field intensity in the capacitor by Gauss' theorem to be $$\mathbf{E} = -\frac{\lambda}{2\pi\varepsilon r}\mathbf{e}_r = \frac{-Q}{2\pi\varepsilon L r}\mathbf{e}_r.$$ The potential difference between the inner and outer capacitors is $$V = -\int_a^b \mathrm{E} \cdot d\mathrm{r} = \frac{\lambda}{2\pi\varepsilon}\ln\left(\frac{b}{a}\right).$$ Hence the capacitance is $$C = \frac{\lambda L}{V} = \frac{2\pi\varepsilon L}{\ln\left(\frac{b}{a}\right)}$$
`gpt-3.5 -turbo` Solution	The capacitance of a cylindrical capacitor is given by the equation: $$C = \frac{2\pi\varepsilon L}{\ln(b/a)}$$ where ε is the dielectric constant, L is the length of the capacitor, a is the radius of the inner conductor, and b is the radius of the outer conductor. Therefore, the capacitance of the given cylindrical capacitor is: $$C = \frac{2\pi\varepsilon L}{\ln(b/a)} = \frac{2\pi\varepsilon L}{\ln(b) - \ln(a)}$$ ANSWER: $\frac{2\pi\varepsilon L}{\ln(b)-\ln(a)}$

Table 13 (continued)

`gpt-4` Solution

To find the capacitance of the cylindrical capacitor, we will use the general formula for capacitance: $C = \frac{Q}{V}$, where Q is the charge stored in the capacitor and V is the potential difference between the inner conductor wire and the outer conducting shell.
First, let's find the electric field E between the two conductors. We can use Gauss's law, which states that the electric flux through a closed surface is equal to the charge enclosed by the surface divided by the permittivity of free space ε_0:

$$\oint \vec{E} \cdot d\vec{A} = \frac{Q_{enclosed}}{\varepsilon_0}.$$

We will choose a cylindrical Gaussian surface of radius r and length L, where $a < r < b$. The electric field is radial, so the electric flux through the curved surface of the cylinder is:

$$\oint \vec{E} \cdot d\vec{A} = E \oint dA = E(2\pi r L).$$

The charge enclosed by the Gaussian surface is the charge on the inner conductor wire, which is given by its linear charge density λ multiplied by the length of the Gaussian surface, L: $Q_{enclosed} = \lambda L$.
Now we can use Gauss's law to find the electric field:

$$E(2\pi r L) = \frac{\lambda L}{\varepsilon_0} \Rightarrow E = \frac{\lambda}{2\pi r \varepsilon_0}.$$

However, we need to account for the dielectric material in between the conductors. The electric field in a dielectric material is related to the electric field in vacuum by the dielectric constant ε:

$$E_{dielectric} = \frac{E}{\varepsilon}.$$

So, the electric field in the dielectric material is:

$$E_{dielectric} = \frac{\lambda}{2\pi r \varepsilon_0 \varepsilon}.$$

Now we can find the potential difference between the inner conductor wire and the outer conducting shell by integrating the electric field along the radial direction:

$$V = -\int_a^b E_{dielectric} dr = -\int_a^b \frac{\lambda}{2\pi r \varepsilon_0 \varepsilon} dr.$$

Integrating, we get:

$$V = -\frac{\lambda}{2\pi \varepsilon_0 \varepsilon} \ln(\frac{b}{a}).$$

Finally, we can find the capacitance using the formula $C = \frac{Q}{V}$:

$$C = \frac{\lambda L}{-\frac{\lambda}{2\pi \varepsilon_0 \varepsilon} \ln(\frac{b}{a})}.$$

Simplifying, we get:
ANSWER: $C = \frac{2\pi \varepsilon_0 \varepsilon L}{\ln(\frac{b}{a})}$.

Table 14: BLEU scores for model recall of the problem statements *when prompted with the first half of the statement*, across different models and parts of ARB. The "Full" score uses the whole model output to compute the BLEU score. We provide two other scores because model outputs tend to be longer than the ground truth and often contain parts of the solution to problems, in addition to the problem statement completion. The "Prefix" score compares the correct problem statement with the model output truncated to the same length as the original problem statement. The "Newlines" score is given by truncating the model output at the first occurrence of the "\n\n" character, because models often continue with redundant information after ending the problem statement with newlines. The three methods produce very similar results, with average scores low, and a rare high top score.

Dataset	Model	Full		Prefix		Newlines	
		Avg	Top	Avg	Top	Avg	Top
Law MCQ	text-davinci-003	0.01	0.12	0.02	0.12	0.00	0.01
	gpt-3.5-turbo-0301	0.02	0.12	0.02	0.21	0.01	0.21
	gpt-4-0314	0.01	0.07	0.01	0.08	0.01	0.07
	claude-v1.3-100k	0.00	0.07	0.00	0.07	0.00	0.00
	gpt-3.5-turbo-instruct	0.01	0.07	0.02	0.11	0.01	0.05
MCAT Reading	text-davinci-003	0.01	0.04	0.01	0.04	0.00	0.00
	gpt-3.5-turbo-0301	0.00	0.00	0.00	0.00	0.00	0.00
	gpt-4-0314	0.01	0.03	0.01	0.03	0.00	0.00
	claude-v1.3-100k	0.00	0.00	0.00	0.00	0.00	0.00
	gpt-3.5-turbo-instruct	0.00	0.02	0.00	0.02	0.00	0.02
MCAT Science	text-davinci-003	0.04	0.12	0.05	0.17	0.00	0.04
	gpt-3.5-turbo-0301	0.03	0.10	0.04	0.15	0.02	0.15
	gpt-4-0314	0.05	0.16	0.06	0.24	0.06	0.24
	claude-v1.3-100k	0.00	0.07	0.00	0.07	0.00	0.00
	gpt-3.5-turbo-instruct	0.04	0.12	0.05	0.17	0.02	0.09
Math Numerical	text-davinci-003	0.05	0.60	0.07	0.85	0.06	0.85
	gpt-3.5-turbo-0301	0.01	0.09	0.01	0.25	0.01	0.16
	gpt-4-0314	0.01	0.08	0.02	0.21	0.01	0.15
	claude-v1.3-100k	0.01	0.09	0.00	0.07	0.00	0.05
	gpt-3.5-turbo-instruct	0.04	0.56	0.06	0.72	0.06	0.72
Math Symbolic	text-davinci-003	0.00	0.00	0.00	0.00	0.00	0.00
	gpt-3.5-turbo-0301	0.00	0.05	0.01	0.15	0.01	0.14
	gpt-4-0314	0.01	0.05	0.02	0.14	0.01	0.08
	claude-v1.3-100k	0.00	0.00	0.00	0.00	0.00	0.00
	gpt-3.5-turbo-instruct	0.00	0.00	0.00	0.00	0.00	0.00
Math Proof-like	text-davinci-003	0.02	0.21	0.03	0.24	0.00	0.00
	gpt-3.5-turbo-0301	0.00	0.08	0.01	0.18	0.01	0.10
	gpt-4-0314	0.01	0.10	0.01	0.14	0.01	0.10
	claude-v1.3-100k	0.00	0.09	0.00	0.09	0.00	0.00
	gpt-3.5-turbo-instruct	0.01	0.13	0.02	0.16	0.01	0.12
Physics Numerical	text-davinci-003	0.01	0.09	0.02	0.17	0.01	0.11
	gpt-3.5-turbo-0301	0.01	0.10	0.02	0.27	0.02	0.26
	gpt-4-0314	0.03	0.19	0.06	0.45	0.05	0.45
	claude-v1.3-100k	0.01	0.09	0.02	0.14	0.00	0.00
	gpt-3.5-turbo-instruct	0.02	0.12	0.04	0.43	0.04	0.36
Physics Symbolic	text-davinci-003	0.01	0.11	0.01	0.12	0.00	0.01
	gpt-3.5-turbo-0301	0.01	0.08	0.01	0.08	0.00	0.07
	gpt-4-0314	0.01	0.08	0.02	0.12	0.02	0.13
	claude-v1.3-100k	0.00	0.05	0.00	0.05	0.00	0.00
	gpt-3.5-turbo-instruct	0.01	0.07	0.01	0.12	0.01	0.10

Table 15: Prompt used for multiple-choice MCAT and Law problems.

System	You are a top graduate student taking an open-ended qualifying exam. Your final answer should always be in the last line of your response, preceded by ANSWER:.
User	You are a top graduate student taking a qualifying exam. Below you will find a multiple choice question. Question: *{Problem_Statement}* Answer Choices: *{Answer_Choices}* Now it is time to choose an answer. Think carefully and go step by step. Make sure to justify all your work. Your final answer should be one of A,B,C,D,... given at the end of your work and preceded by ANSWER:. For example, if you think the answer is B, the last line of your answer should be ANSWER: B Solution:

Table 16: Prompt used for numerical problems.

System	You are a top graduate student taking an open-ended qualifying exam. Your final answer should always be in the last line of your response, preceded by ANSWER:.
User	You are a top graduate student taking an open-ended qualifying exam. Below you will find a question requiring you to compute a numerical value. Question: *{Problem_Statement}* Now it is time to give your answer. Think carefully and go step by step. Make sure to justify all your work. Please simplify all expressions as much as possible and do not leave any variables in your final answer. Your final answer should NOT contain units and should be given at the end of your work and preceded by ANSWER: For example, if you think the answer is 2.4 meters, the last line of your answer should be ANSWER: 2.4. Solution:

Table 17: Prompt used for symbolic problems.

System	You are a top graduate student taking an open-ended qualifying exam. Your final answer should always be in the last line of your response, preceded by ANSWER:.
User	You are a top graduate student taking an open-ended qualifying exam. Below you will find a question requiring you to give a symbolic answer. Question: *{Problem_Statement}* Now it is time to give your answer. Think carefully and go step by step. Make sure to justify all your work. Your final answer should NOT contain units and should be given at the end of your work and preceded by ANSWER: For example, if you think the answer is $x * y$, the last line of your answer should be ANSWER: $x * y$ Solution:

Table 18: Prompt used for proof-like problems.

System	You are a top graduate student taking an open-ended qualifying exam. Your final answer should always be in the last line of your response, preceded by ANSWER:.
User	You are a top graduate student taking an open-ended qualifying exam. Below you will find a question requiring you to prove the given statement. Question: *{Problem_Statement}* Now it is time to give your answer. Think carefully and go step by step. Make sure to justify all your work. Solution:

Table 19: Prompt used for GPT-3.5-turbo symbolic evaluation.

System	You are a top professor grading an open-ended qualifying exam.
User	Problem Statement: Give a solution to the differential equation $y'' = -y$ Reference Answer: $y(t) = cos(t)$ or $y(t) = sin(t)$ Model Answer: $y(x) = sin(x)$ Score: The correct answer is either $y(t) = cos(t)$ or $y(t) = sin(t)$. The model gave $y(x) = sin(x)$. Since the function variable was not specified, the model matches one of the reference answers. GRADE: 1 Problem Statement: *{Problem_Statement}* Reference Answer: *{Final_Answer}* Model Answer: *{model_final_answer}* Now it is time to grade the model answer. If the solution is incorrect give GRADE: 0. If the solution is nearly correct up to a constant give GRADE: 0.5. If the solution is correct give GRADE: 1. Before coming to a final grade think think carefully and go step by step. DO NOT TRY TO SOLVE THE PROBLEM. If a variable name is not specified and the reference answer and the model answer are the same up to the name of a variable give a score of GRADE: 1. For example if the reference answer is $f(x) = x^2$ and the model answer is $f(y) = y^2$ give a score of GRADE: 1. Score: The reference answer

Table 20: Prompt used for GPT-4 rubric self-evaluation.

System	You are a top professor grading an open-ended qualifying exam.

User	Problem Statement: *{Problem_Statement}*
	Rubric: *{rubric}*
	Student Answer: *{response}*
	
	Now it is time to grade the student answer. Make sure to check each point of the rubric step by step. And make sure to print the total number of earned points at the end of your grading. For example, if the student earned 8 points, print Rubric Score: 8 points
	
	Rubric Evaluation:

Table 21: Prompt used for GPT-4 rubric design.

System	You are a top professor grading an open-ended qualifying exam.

User	Problem Statement: *{Sample_Problem_Statement}*
	Reference Solution: *{Sample_Reference_Solution}*
	Rubric: *{Handcrafted_Rubric}*
	
	Problem Statement: *{ Problem_Statement }*
	Reference Solution: *{ Solution }*
	
	Now it is time to write the rubric. Make sure to think carefully and go step by step, breaking down the problem into multiple parts. The total number of possible points should sum to 10.
	
	Rubric:

Table 22: Example model generated rubric and evaluation. Evaluation with a rubric reveals that the model begins with the model makes partial progress, but is unable to arrive at the correct final answer.

<table>
<tr>
<td>Problem Statement</td>
<td>Find the maximum area of all triangles that can be inscribed in an ellipse with semiaxes a and b, and describe the triangles that have maximum area.</td>
</tr>
<tr>
<td>Ground Truth Solution</td>
<td>

Using the parameterization

$$x = a\cos t, y = b\sin t,$$

a triple of points on the ellipse is given by

$$\left(a\cos t_i, b\sin t_i\right), \quad i = 1,2,3.$$

So the area of an inscribed triangle is given by

$$\frac{1}{2}\left|\begin{array}{ccc} 1 & a\cos t_1 & b\sin t_1 \\ 1 & a\cos t_2 & b\sin t_2 \\ 1 & a\cos t_3 & b\sin t_3 \end{array}\right| = \frac{ab}{2}\left|\begin{array}{ccc} 1 & \cos t_1 & \sin t_1 \\ 1 & \cos t_2 & \sin t_2 \\ 1 & \cos t_3 & \sin t_3 \end{array}\right|$$

which is ab times the area of a triangle inscribed in the unit circle. In the case of the circle, among all inscribed triangles with a given base $2w(0 < w \leqslant 1)$, the one of maximum area is an isosceles triangle whose area equals

$$g(w) = w\left(1 + \sqrt{1 - w^2}\right).$$

Using elementary calculus one finds that the maximum of g on the interval $0 \leqslant w \leqslant 1$ occurs at $w = \sqrt{3}/2$, corresponding to an equilateral triangle, and equals $3\sqrt{3}/4$. Alternatively, fixing one side of the triangle as the basis, we easily see that among all the inscribed triangles the one with the greatest area is isosceles because of the maximum height, showing that the angle at the basis is the same. Fixing another side we see that the triangle is indeed equilateral. Hence, the area is maximal when

$$t_2 = t_1 + \frac{2\pi}{3} \text{ and } t_3 = t_2 + \frac{2\pi}{3}$$

that is, when the corresponding triangle inscribed in the unit circle is regular.
For the ellipse with semiaxes a, b, this corresponds to an inscribed triangle with maximum area equals $3ab\sqrt{3}/4$.

</td>
</tr>
<tr>
<td>Model Generated Rubric</td>
<td>

- Parameterize the ellipse using $x = a\cos t, y = b\sin t$ (2 points).
- Write the expression for the area of an inscribed triangle using the determinant formula (2 points):

$$\frac{ab}{2}\left|\begin{array}{ccc} 1 & \cos t_1 & \sin t_1 \\ 1 & \cos t_2 & \sin t_2 \\ 1 & \cos t_3 & \sin t_3 \end{array}\right| \tag{1}$$

- Connect the problem to the maximum area of an inscribed triangle in the unit circle (2 points).
- Correctly identify that the maximum area occurs when the inscribed triangle is equilateral (2 points).
- Find the maximum area of an inscribed triangle in the ellipse, and provide the correct answer (2 points):

$$\frac{3ab\sqrt{3}}{4} \tag{2}$$

</td>
</tr>
</table>

Table 22 (continued)

Model Response	To maximize the area of the inscribed triangle, we can use the following approach: 1. Parameterize the ellipse: The equation of an ellipse with semiaxes a and b is given by $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. We can parameterize the ellipse using the parameter t as follows: $x = a\cos(t)$ and $y = b\sin(t)$, where $0 \leq t \leq 2\pi$. 2. Determine the vertices of the triangle: Let $A(a\cos(t_1), b\sin(t_1))$, $B(a\cos(t_2), b\sin(t_2))$, and $C(a\cos(t_3), b\sin(t_3))$ be the vertices of the inscribed triangle, where $0 \leq t_1, t_2, t_3 \leq 2\pi$. 3. Calculate the area of the triangle: Using the Shoelace Theorem, the area of the triangle is given by: $$\text{Area} = \frac{1}{2}\left\|\sum_{i=1}^{3} x_i y_{i+1} - \sum_{i=1}^{3} x_{i+1} y_i\right\|$$ where $(x_4, y_4) = (x_1, y_1)$. Substituting the parameterized coordinates, we get: $$\text{Area} = \frac{1}{2}\|a^2\cos(t_1)\sin(t_2) + a^2\cos(t_2)\sin(t_3) \\ +a^2\cos(t_3)\sin(t_1) - a^2\cos(t_2)\sin(t_1) - a^2\cos(t_3)\sin(t_2) - a^2\cos(t_1)\sin(t_3)\|$$ Simplifying the expression, we get: $$\text{Area} = \frac{a^2}{2}\|\cos(t_1)\sin(t_2) - \cos(t_2)\sin(t_1) + \\ \cos(t_2)\sin(t_3) - \cos(t_3)\sin(t_2) + \cos(t_3)\sin(t_1) - \cos(t_1)\sin(t_3)\|$$ Using the sine difference identity, $\sin(A-B) = \sin(A)\cos(B) - \cos(A)\sin(B)$, we can rewrite the area as: $$\text{Area} = \frac{a^2}{2}\left\|\sin(t_1 - t_2) + \sin(t_2 - t_3) + \sin(t_3 - t_1)\right\|$$ 4. Maximize the area: Notice that the maximum value of each sine term is 1, and this occurs when the angles t_1, t_2, t_3 are equally spaced around the ellipse, i.e., when the triangle is equilateral. In this case, the area becomes: $$\text{Area} = \frac{a^2}{2}\left\|1 + 1 + 1\right\| = \frac{3a^2}{2}$$ 5. Relate the maximum area to the semiaxes: Since the triangle is equilateral, the ratio of the area of the triangle to the area of the ellipse is constant. The area of the ellipse is given by πab. Thus, we can write: $$\frac{\text{Area}_{\text{max}}}{\pi ab} = \frac{\frac{3a^2}{2}}{\pi ab}$$ Solving for the maximum area, we get: $$\text{Area}_{\text{max}} = \frac{3a^2}{2} \cdot \pi ab = \frac{3\pi a^3 b}{2}$$ Thus, the maximum area of an inscribed triangle in an ellipse with semiaxes a and b is $\frac{3\pi a^3 b}{2}$, and the triangles with maximum area are equilateral. ANSWER: $\frac{3\pi a^3 b}{2}$

Table 22 (continued)

Rubric Based Evaluation	1. The student parameterized the ellipse correctly using $x = a\cos(t)$ and $y = b\sin(t)$. (2 points) 2. The student used the Shoelace Theorem to find the area of the triangle, but did not use the determinant formula as specified in the rubric. (0 points) 3. The student did not explicitly connect the problem to the maximum area of an inscribed triangle in the unit circle. (0 points) 4. The student correctly identified that the maximum area occurs when the inscribed triangle is equilateral. (2 points) 5. The student found the maximum area of the inscribed triangle in the ellipse, but their answer is incorrect. The correct answer is $\frac{3ab\sqrt{3}}{4}$, not $\frac{3\pi a^3 b}{2}$. (0 points) Rubric Score: 4 points

Table 23: Performance of GPT-3.5-turbo on symbolic equivalence versus human ground truth. The model achieves a false positive rate of 0, but allows many false negatives. This suggests there are many cases in which the model does not recognize equivalent expressions.

	Physics Symbolic	Math Symbolic
True positives	3	4
False positives	0	0
True negatives	9	22
False negatives	6	8
Accuracy	67%	76%

Table 24: One-shot prompt used for numerical problems, example before question.

System	You are a top graduate student taking an open-ended qualifying exam. Your final answer should always be in the last line of your response, preceded by ANSWER:.
User	You are a top graduate student taking an open-ended qualifying exam. Below you will find a question requiring you to compute a numerical value. Example Question: *{Example_Problem_Statement}* Now it is time to give your answer. Think carefully and go step by step. Make sure to justify all your work. Please simplify all expressions as much as possible and do not leave any variables in your final answer. Your final answer should NOT contain units and should be given at the end of your work and preceded by ANSWER: For example, if you think the answer is 2.4 meters, the last line of your answer should be ANSWER: 2.4. Example Solution: *{Example_Full_Solution}* Example Final Answer: ANSWER: *{Example_Final_Answer}* Question: *{Problem_Statement}* Now it is time to give your answer. Think carefully and go step by step. Make sure to justify all your work. Please simplify all expressions as much as possible and do not leave any variables in your final answer. Your final answer should NOT contain units and should be given at the end of your work and preceded by ANSWER: For example, if you think the answer is 2.4 meters, the last line of your answer should be ANSWER: 2.4. Solution:

Table 25: Scores for sampled Math and Physics Numerical, one-shot prompting method in Table 24.

	Math Numerical	Physics Numerical
`gpt-4-0314`	16%	22%
`gpt-3.5-turbo-0301`	10%	7%
`text-davinci-003`	10%	6%
`claude-v1.3-100k`	6%	11%