

Mathematical Capabilities of ChatGPT

S. Frieder^{*1}, L. Pinchetti¹, R. Griffiths³, T. Salvatori², T. Lukasiewicz^{1,2}, P. Petersen^{4,5}, A. Chevalier⁶, and J. Berner⁴

¹Department of Computer Science, University of Oxford, UK

²Institute of Logic and Computation, TU Wien, Austria

³Department of Physics University of Cambridge, UK

⁴Faculty of Mathematics, University of Vienna, Austria

⁵Research Network Data Science, University of Vienna, Austria

⁶School of Mathematics, Institute for Advanced Study, US

January 27, 2023

Abstract

We investigate the mathematical capabilities of ChatGPT by testing it on publicly available datasets, as well as hand-crafted ones, and measure its performance against other models trained on providing mathematical answers, such as Minerva. We also test whether ChatGPT can be a useful assistant to professional mathematicians by emulating various use cases that come up in daily professional activities of mathematicians (question answering, theorem searching). To date, the existing natural language mathematical datasets used to benchmark language models only cover elementary math. We address this issue by introducing a much more sophisticated mathematics dataset: GOATSS. This new dataset is the first natural-language dataset made and curated by working math researchers, that aims to provide a holistic overview of the mathematical capabilities of language models. We benchmark ChatGPT on GOATSS and evaluate performance using highly fine-grained mathematical criteria. We make this new dataset publicly available¹ in order to assist a community-driven comparison of ChatGPT with (future) large language models in terms of advanced mathematical comprehension. The main conclusion is that, contrary to many positive reports in the media (a potential case of selection bias), ChatGPT still suffers from severe performance issues on graduate-level mathematics.

1 Introduction

Since its introduction, ChatGPT has rapidly become a widely known question-and-answer dialogue system. It has been mentioned in traditional media across the globe [31; 24; 19] and across all major internet platforms [28; 37; 11]. According to Twitter data, it is by far the most talked about language model to date; cf. Figure 1.

The performance of ChatGPT has been analyzed in a large number of exam-related use cases, with varying degrees of scientific rigor, ranging from detailed studies to anecdotal evidence. Use cases include passing the United States Medical Licensing Examination [15], scoring highly on the Psychology Today Verbal-Linguistic Intelligence IQ Test [32], and answering (and generating) Operations Management exam questions that were deemed to be within the scope of a typical MBA curriculum [38], all with a performance that elicited positive surprise of the authors. Due to this and other reasons, it is widely believed that large language models (LLMs) will impact a large number of areas and will be used as assistants by many professionals.

^{*}Corresponding author: simon.frieder@wolfson.ox.ac.uk

¹<https://github.com/Fr-S/GOATSS>

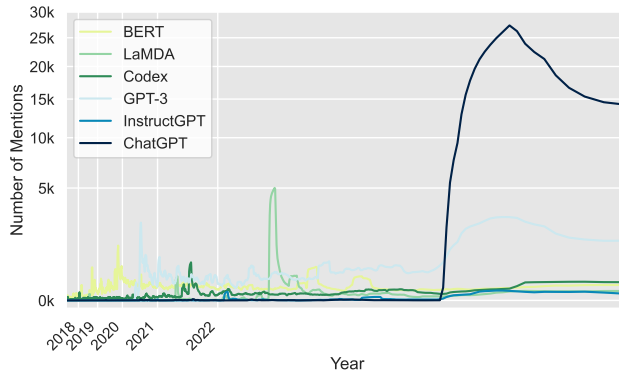


Figure 1: Log-plot with Twitter count data relating the counts of a selection of notable large language models from the beginning of the release date of GPT-3. ChatGPT counts by far dominate those of all other language models. Vertical year-ticks denote the end of the mentioned year.

In this article, we will focus on performing a detailed analysis of the mathematical capabilities of ChatGPT. This includes, but is not limited to, answering exam-style mathematical questions and investigating how ChatGPT behaves in a number of mathematical contexts. Our analysis includes testing how many of the skills ChatGPT can emulate that are necessary to do professional mathematics. Examples of such skills are the ability to answer computational questions (“What is the integral of $\int_0^{\frac{\pi}{2}} \arccos(\frac{\cos x}{1+2\cos x}) dx$?”), the ability to complete mathematical proofs that have gaps or missing steps, the ability to solve questions that are more focused on deep insights and original solutions, such as those of mathematical olympiads, and the ability to survey the literature and think across domains (“which other theorems do we need to prove a given theorem?”).

To do this, we have designed a thorough testing methodology to evaluate the outputs of ChatGPT, including error codes that represent various possible behaviors of ChatGPT (see Section 3). We score ChatGPT’s responses, we report on the results using this methodology, and we compare ChatGPT to the state-of-the-art models trained for mathematical comprehension. In addition, we have created new datasets of prompts that are aimed to test specific aspects of ChatGPT related to mathematical comprehension.

Furthermore, we evaluate ChatGPT by comparing it to random samples from existing datasets that were devised to test models that were specifically trained for mathematical comprehension [16; 14]. A number of the datasets are specifically designed so that the questions could not be answered if ChatGPT were memorizing the results. All of those datasets are created by the authors.

We note that we focus on the 9th-January-2023 version of ChatGPT [23], as made available through web access at <https://chat.openai.com/chat>. This is the latest version at the time of writing. Focusing on this version is necessary because precise details of the model architecture and, in particular, ChatGPT’s training methodology have not been released in the introductory statement [35] by its creator, OpenAI. ChatGPT is the latest model of the GPT lineage [23], being based on InstructGPT, which in turn is based on a trained GPT-3 [2], and fine-tuned using reinforcement learning with human feedback [25]. We note that even for models that predate ChatGPT, such as InstructGPT, where research articles and model cards [40] have been released, full reproducibility is not possible since the code and exact datasets have not been released. Furthermore, it was confirmed by OpenAI employees that a slight mismatch exists between the trained model that is accessible via OpenAI web interface and the model referred to in the official paper [36]. This indicates how essential it is to document carefully which model our analysis pertains to; in our dataset, we have included time stamps for each prompt in order to be able to track should any changes in ChatGPTs version that have occurred that may not have been communicated publicly by OpenAI.

The contributions of this article are threefold:

- First, insight for mathematical use is provided. We show for which types of questions and which domains of mathematics, ChatGPT may be useful and how it could be integrated into the workflow of

a mathematician;

- Second, failure modes of ChatGPT are identified, as well as limits of its capabilities. This can aid future efforts to build LLMs that perform better on mathematics. Our analysis is akin to a *mathematical model card*, where the mathematical strong points and weaknesses are summarized (see Section 4).
- Third, we provide benchmarks for testing mathematical capabilities of future LLMs, so that they can be compared to ChatGPT across a range of aspects regarding advanced mathematical comprehension. These benchmarks are derived from the most advanced datasets regarding mathematical queries for language models that exist today. We release this dataset publicly, and we encourage community participation by allowing Github pull requests in order to grow the datasets beyond their current sizes.

2 Related Work

As a large language model, ChatGPT can be universally employed to perform mathematical reasoning and therefore has to compare with technologies that in this space are sometimes decades old. Performing mathematical reasoning in an automated way has a long history and can be traced back to 1959 [34], the most focus being devoted to proving theorems [8]. Presently, there is a realization that the classical approaches, using a symbolic encoding of mathematics, have reached a plateau [12].

There is now a growing body of literature on learning mathematical relationships directly in a supervised-learning manner [1; 7; 13] or by using LLMs to perform mathematical reasoning directly on mathematics encoded in natural language [17]. Sometimes, the distinction is blurred because Transformers can also be used in a supervised-learning setting and have been employed successfully in learning mathematical relationships [16; 3].

Most recently published large language models, such as PaLM [5; 26], released in 2022, are tested only on elementary-level mathematical reasoning datasets. We speculate this is because the obtained results already suggest that the models struggle on much simpler datasets, such as the GSM8K dataset [6]. For example, the version of PaLM with 540 billion parameters with chain-of-thought prompting and access to an external calculator solves only 58% on the GSM8K dataset. This model nonetheless outperforms GPT-3 [2] on the same dataset, which only solves at best 54%; this is consistent with the performance of older models, for example, variations of BERT [27] only solve between 28% and 37% of the problems when fine-tuned and tested on the AQUA-RAT (Algebra Question Answering with Rationales) dataset [18]. In some cases, such as the LaMDA model [39] released also in 2022 by Google, an evaluation of mathematical reasoning is missing entirely.

Among the mentioned LLMs, Minerva [17], based on PaLM, stands out, being trained in equal parts on websites that contain MathJax elements and arXiv preprints (and on general natural language data on which PaLM was trained), achieving a score around 50% on the MATH dataset [14].

Among the purely supervised approaches, we mention [16], where a Transformer architecture was used to generate symbolic solutions to integrating functions and finding closed-form solutions to first-order and second-order differential equations. which outperformed classical solvers, such as Mathematica, MATLAB, and Maple by at least 14% on a test set of integration problems. On the task of solving differential equations, the Transformer-based approach still exceeds the classical approach, but by a smaller margin (at least 4% in case of first-order differential equations, and with more varied results for second-order equations). An up-to-date survey on mathematical datasets and performance of various LLMs can be found in [20].

For ChatGPT so far, most investigations related to mathematical reasoning consist of anecdotal evidence concerning its performance and its failure modes; see, e.g., [11; 30; 21; 37]. Unfortunately, a clear methodology is missing, as most of the results are scattered on various internet platforms and are not easily reproducible.

Our dataset goes beyond all the mentioned mathematical datasets for large language models in terms of mathematical sophistication and in terms of the different aspects of mathematical reasoning that are being tested, including those mentioned in the survey [20]. On the other hand, we like to mention in the case of *formalized* mathematics, there exist datasets that encode advanced mathematical concepts. Some of the ideas that we have used in this article, such as prompting with missing proofs, are echoed in [29]. Yet, for the

Dataset name	Comprised of the file(s)	Tags
<i>Grad-Text</i>	W. Rudin, Functional Analysis (ch. 1)	M3 Q4
	W. Rudin, Functional Analysis (ch. 2)	M3 Q4
	J. Munkres, Topology (ch. 1)	M3 Q4
	J. Munkres, Topology (ch. 2)	M3 Q4
	R. Durrett, Probability Theory	M3 Q4
<i>Olympiad-Problem-Solving</i>	Olympiad Problem Solving	M4 Q4 D2
<i>Aperture-Proofs</i>	Proofs Collection A	M3 Q2 Q5
	Proofs Collection B Prealgebra	M1 Q5
	Proofs Collection B Precalculus	M1 Q5
<i>MATH</i>	MATH Algebra	M1 M2 M3 Q3 Q4
	MATH Counting and Probability	M1 M2 M3 Q3 Q4
	MATH Prealgebra	M1 Q3 Q4
	MATH Precalculus	M1 Q3 Q4
<i>Symbolic-Integration</i>	Symbolic Integration	M2 Q3 D1
<i>Search-Engine-Aspects</i>	Definition Retrieval	M3 Q1 Q2 D3
	Named Theorem Proof Completion	M3 Q1 Q2 D3

Table 1: A summary of all datasets, together with their associated tags. The tags M_i , Q_i , and D_i relate to the level of *Mathematical* difficulty, the *Question* type, and the *Out-of-Distribution* type from Section 3.3, respectively.

purpose of doing mathematics with large language models, these formal datasets cannot be leveraged, since no straightforward way exists to convert them to natural language (in addition to various issues such as bias, that might occur in the context of an automatic conversion).

3 Datasets

3.1 Dataset creation

We assess the mathematical reasoning capabilities of ChatGPT by creating a collection of multiple datasets of prompts, totalling 667 prompts, for which ChatGPT’s output was manually rated by experts. Then, we record and rate each of the outputs provided by the model. The combined effort of devising mathematically insightful prompts, some of which are at graduate-level mathematics, and carefully rating the output of ChatGPT amount to several hundreds of person-hours.

Our collection is comprised of 6 subdatasets (included as folders in the code), called

- ***Grad-Text***
- ***Olympiad-Problem-Solving***
- ***Aperture-Proofs***
- ***MATH***
- ***Symbolic-Integration***
- ***Search-Engine-Aspects***

We summarize those in Table 1. The letters that are set in boldface make up the GOATSS acronym.

Two of the subdatasets, the *MATH* subdataset and the *Symbolic-Integration* subdataset, use prompts taken from existing datasets, [14] and [16], respectively. This was done in order to be able to compare how ChatGPT performs against existing state-of-the-art models, one based on an LLM, Minerva [17], and one based on a supervised-learning approach [16]. This does not mean, that the use of these datasets was

automatic, since in both cases the authors, as experts in the field, rated the output. Furthermore, in the second case, a conversion from Polish notation was necessary.

The other subdatasets were hand-crafted by the authors. We note that it is neither possible to outsource the creation of these datasets to a crowdsourcing service, such as Amazon Mechanical Turk, nor is it possible to generate these datasets automatically from code, because advanced mathematical insight is required for the creation of each prompt (though based on our work, it might be possible to extend the dataset by creating variations of our questions in a purely programmatic manner; see Section 5). This raises the difficulty of the creation of more data, since graduate-level mathematics (and in some cases PhD-level mathematics) is required. The mathematical skill level of the authors ² matches this requirement.

3.2 Format

The format of each of our subdatasets that make up our GOATSS dataset (the acronym is explained in table 1, by following the boldface characters) follows the same convention and each subdataset consists of (potentially multiple) JSON-formatted files with entries such as in the example below. Our format is similar to, e.g., the AQuA-RAT (Algebra Question Answering with Rationales) dataset [18]. A single datapoint in a JSON file has the following form:

```
{
  "prompt": "Can you quote a famous mathematical
            theorem?",
  "output": "\"Euclid's Elements, Proposition 47:
            In right-angled triangles, the
            square on the side opposite the
            right angle is equal to the sum of
            the squares on the other two
            sides.\"\"",
  "rating": "5",
  "errorcodes": [],
  "warningcodes": ["w1"],
  "comment": "This is actually Pythagoras'
            theorem, which was not mentioned,
            hence only a warning flag is
            raised.",
  "ref" : "",
  "msc": "51M04",
  "confidence": "high",
  "timestamp" : ">9_Jan_2023"
}
```

We explain each of these fields; unless explicitly mentioned below, all fields need to have a value:

- **prompt** denotes the input that we supply to ChatGPT via its web interface. We always use a new session for each prompt to avoid biasing ChatGPT by using previous prompts. This is consistent with how a mathematician might use ChatGPT: they might want to get the right answer on the first try, not after a time-consuming interaction during which ChatGPT would be able to tailor its responses.
- **output** denotes the raw output that ChatGPT supplies us with, including escape characters, such as `\` or `\n`;
- **rating** is a number from 1 to 5 that shows how many points ChatGPT has scored, 5 being a perfect answer, and 1 being the lowest. A detailed explanation regarding the rating policy that we followed is contained in the appendix, Section A.

²Due to anonymization, we cannot give an exact description at this stage, but we will include background information in case of acceptance.

- **errorcodes** highlight a list of error types, that illustrate the failure mode of ChatGPT in a more fine-grained way. Not all types of errors apply to all (sub)datasets: For example, an error code for a missing proof step would not be applicable on a dataset that tests whether ChatGPT can multiply numbers or find prime divisors. This field can be empty, if no error code applies. The detailed explanation of the error codes (and the warning codes; see below) that was provided to the annotators is contained in the appendix, Section B.
- **warningcodes** highlight any problematic aspects of ChatGPT; for example, ChatGPT might be rambling and providing you with unrelated information, or use a poor (but correct) way of solving problems. This field can be empty, if no warning code applies.
- **comment** denotes any noteworthy commentary that an assessor of ChatGPTs may make. This can be related to giving a more detailed explanation of output (or its failure modes), providing reasoning behind awarding a certain error code, generally providing context etc. For some subdatasets (see Section 3.3), this field was used to indicate the difficulty level of the prompt, as well as an official solution, if available. This field is not required to always have a value.
- **msc** denotes the math subject classification³ (MSC) that pertains to the output, not the prompt that one gives to ChatGPT. This is because the prompt, unlike the output, might not really have a classification, for example, if ChatGPT is asked what the most important theorem in all of math is⁴.
- **ref** indicates a reference to where the prompt was originally taken from (for some subdatasets, such as *Aperture-Proofs*, we have changed proofs from various books or `math.stackexchange.com`; the original source was recorded in this field). This field can be empty, if the question was formulated by the authors, and no authoritative source was plausible.
- **confidence** indicates how confident we have perceived ChatGPT to be when presenting us with its output. It is usually very confident, unlike other GPT-like models such Codex, that we have experimented with. Consider in this sense the following statement (of statements):

Here are two statements: "Let (X, A, μ) be a measure space. Show that the signed measures $\lambda: A \rightarrow \mathbb{R}$ form a Banach space."; "Let (X, A, μ) be a measure space. Show that the signed measures $\lambda: A \rightarrow \mathbb{R}$ form a Banach space." Are they equivalent?

The response by ChatGPT is not phrased in order to show any nuance in terms of confidence (this is typical, even if ChatGPT is wrong):

Yes, the two statements are equivalent. They both express the same mathematical idea, but use different notation and formatting conventions. The first statement uses parentheses to enclose the elements of the measure space and regular letters to represent the functions, while the second statement uses subscripted parentheses and uses the LaTeX formatting to represent the elements of the measure space and the function.

³A complete list of MSC codes can be accessed under the URL: <https://zbmath.org/static/msc2020.pdf>

⁴The answer is Pythagoras' theorem, according to ChatGPT.

The response by Codex [4], another model that we briefly tested (but whose scope would have exceeded that of a single conference article) gives a cautious response and, unlike ChatGPT, is capable of voicing doubt:

I think they are equivalent, but I am not sure

Because we aim for our dataset to be future-proof, we have decided to keep the `confidence` field, even if for ChatGPT there is no immediate information to be gained by collecting this piece of information.

- The `timestamp` denotes when the prompt was entered into ChatGPT.

Each subdataset is made up of multiple such data points. In cases where we used prompt engineering, and asked variations of a single question, we enclosed multiple such data points in square brackets.

The fields within a single data point interact in nontrivial ways: If a rating of 5 is given, then it is expected that no error code is present—though there may be warning codes that are used. The error codes and warning codes are loosely in the spirit of a compiler throwing errors and warnings if it is given incorrect or sloppy code—although we have a role reversal, where the human is now the compiler and the machine the code produced. In this sense, for some prompts, we have used multiple error and/or warning codes, which is why these fields are lists of strings. We use these codes to collect statistics on the behavior of ChatGPT; see Section 4.

The usage of MSC codes can be useful for mathematicians who want to integrate ChatGPT in their daily workflow, as it allows to know in which areas the model performs better and can hence be trusted more. Our dataset is very diverse, as it has a total 41 MSC codes, the top occurring being 00A07 (124 occurrences), followed by 26A06 (61 occurrences) and 11A05 (47 occurrences). An exhaustive survey of ChatGPT’s performance across *every* MSC code would necessitate a large, community-driven effort to set up a very large database. Because of the high cost of rating each output, which requires specialized skills, this is something that no individual research group could reasonably do—but we hope that our approach is a starting point for such an effort.

For end-users of ChatGPT, it is desirable to avoid having a long-winded dialogue to arrive at a solution. Therefore, we require that ChatGPT gives us the correct solution by providing it only the input, without any subsequent clarification. All chats with ChatGPT are thus “cold”. But we do allow the possibility of prompt engineering of the input, where more information is added beyond the core prompt content.

3.3 The subdatasets

Some of the subdatasets contain prompts that may be protected under copyright (e.g., exercises from textbooks). In these cases, we have used the SHA256 hash function, and the publicly released datasets will only contain the hashed value of the prompt that we used, together with a detailed reference that points to the exact text page and theorem or exercise that was used. The non-hashed version of these datasets are kept privately. For most of our subdatasets, we have used \LaTeX to encode mathematical input. Our experiments have shown that ChatGPT can process \LaTeX -encoded mathematics well. For example, on the *Aperture-Proofs* dataset, except for one case, the output of the prompts by ChatGPT was valid \LaTeX source code and could be rendered.

The *Grad-Text* subdataset consists of a collection of books that used widely in universities to teach upper undergraduate or first-year graduate courses in a degree in mathematics. We have used as prompt and rated the output most of the exercises from the first and second chapters of these books (excepting the book by Durrett [9] that only contains the first chapter).

The *Olympiad-Problem-Solving* subdataset consists of a selection of exercises from the book *Problem Solving Strategies* by A. Engel [10], that is often used when preparing for mathematical competitions. We selected and graded ChatGPT output on one hundred exercises, drawn from all chapters.

The *Aperture-Proofs* subdataset consist of a number of proofs, drawn from `math.stackexchange.book`, a collection of books and the MATH dataset [14] where intentionally parts of the proofs were deleted and

ChatGPT was prompted to fill the gaps: This was done either by using a MISSING token, or by finishing the proof early and prompting ChatGPT to complete it.

The *MATH* subdataset consists of a random sample prompts from the MATH dataset [14]. The MATH dataset by [14] comes with a Level of difficulty of each problem. We focused our random samples on two domains, Algebra and Probability Theory, but sampled the same number of problems at each level of difficulty.

The *Symbolic-Integration* subdataset consist of a random samples of integrals that were in the test set of [16]. We converted these from Polish notation to L^AT_EX before prompting ChatGPT. Assessment was done by comparing with a classical symbolic integration system, Mathematica.

The *Search-Engine-Aspects* subdataset consists of problems that were not sampled from a particular source and generated by an human expert in the field. In the file *Named Theorem Proof Completion* we focused on prompting ChatGPT to provide proof outlines of various theorems that are sufficiently well known within Functional Analysis to have names. In the *Definition Retrieval* file we prompted ChatGPT to state correctly various definitions centered around Functional Analysis and Topology. This subdataset has a strong oral character, as prompt phrased in this way are unlikely to be found in many written mathematical texts.

Our subdatasets can be categorized along multiple dimensions, which we state below:

Mathematical difficulty (ascending):

1. elementary arithmetic problems, as found in the MATH dataset [14] at lower levels of difficulty;
2. symbolic problems (integration of functions) that can be also solved via a supervised-learning, data-driven approach to mathematics [16];
3. graduate-level or upper-undergraduate exercises from well-known textbooks [33; 22; 9], spanning diverse domains, from topology and probability theory, as well as from `math.stackexchange.com`;
4. exercises that are in the style of mathematical olympiad problems, such as those taken from Engel’s *Problem Solving Strategies* book [10];

Question type:

1. Review questions, which ask to state or name correctly certain mathematical facts (the *Definition Retrieval* file of the *Search-Engine-Aspects* subdataset);
2. Overview-type review questions, which cut through an entire field of mathematics (the *Named Proof Completion* part of the *Search-Engine-Aspects* subdataset, as well the *Aperture-Proof* subdataset);
3. Computational questions (the *Symbolic Integration* subdataset as well as various prompts from the *MATH* dataset);
4. Proof-based questions, which ask for a theorem proof or for a puzzle solution (The *Olympiad-Problem-Solving* subdataset, the *Grad-Text* subdataset);
5. Proof-completion questions, which ask for a proof that has missing gaps, or is incomplete, to be completed (the *Aperture-Proof* subdataset as well as various prompts from the MATH dataset)

Types of high out-of-distribution likelihood:

1. Nontrivial problem encoding: The data points from the *Symbolic Integration* subdataset come from [16] and are publicly available⁵. Since the online training set uses Polish notation, it is very unlikely that ChatGPT has seen these exact prompts before;
2. Succinct solution: The solution for the *Olympiad-Problem-Solving* subdataset are included in the book by Engel ([10]). But the solutions are extremely concise, and simply repeating them would not show an immediate understanding of the problem;

⁵<https://github.com/facebookresearch/SymbolicMathematics>

3. Spoken dialogue: The *Search-Engine-Aspects* subdataset is unlikely to be well represented in the data on which ChatGPT has been trained, since its prompts resemble word fragments that might appear in a mathematical dialogue (e.g., an oral mathematical exam), rather than in a textbook;

One could in theory start to investigate every possible combination of these attributes (e.g., for elementary arithmetic problems, in a non-trivial encoding, one could generate data to cover every possible question type listed above). This would lead to 60 (sub)datasets, which, because of the manual, skilled curation effort, is by far too much for a single research group. Hence, we will allow pull requests in order to encourage the community to contribute and grow these datasets, so that they can be used as a useful benchmark for other LLMs. We have nonetheless striven in this work to cover each of these aspects individually, in some dataset, as can be seen in Table 1. Investigating every possible combination and subcombination of attributes of one dataset is not feasible for a single research.

Existing datasets do not cover all possible combinations of possibilities across all these dimensions. Devising further datasets to do so is not feasible. Instead, we will devise specific, well-crafted dataset to cover a number of illustrative cases, which we describe below, so that we cover all these dimensions individually. We collect statistics for ChatGPT’s output, including output length, the stability of the answer under prompt engineering, as well as a personal rating of how close we perceived ChatGPT to be to the correct answer. We also take notice whether the output takes an excessively long time.

Because input to ChatGPT is purely textual, certain types of questions that have a strong geometrical flavour and might be stated and solved in non-text based fashion (e.g., questions involving diagrams or small algorithm, as occasionally occur in [10]) have been excluded.

4 Results

Will ChatGPT get you through a university math class? No, you are better off copying from your average peer! In this section we analyze why, by going through the common mistakes performed by ChatGPT one dataset at the time.

Grad-Text ChatGPT performed best on easy topology questions (the first chapter from the book *Topology* by J. Munkres ([22])), which is reflected in its rating; see Figure 2. On the rest of the books it performed comparatively. We note that it never failed to understand a query, the lowest grade it received is a 2.

Olympiad-Problem-Solving On this subdataset, ChatGPT performed poorly. No rating of 5 was awarded, and only two ratings of 4 were achieved. From a mathematical point of view, these questions were also by far the hardest, as they can also pose difficulty to professional mathematician.

Aperture-Proofs ChatGPT correctly recognized most well-known results or concepts (e.g., filling the mean-value theorem, given a proof where a reference to this theorem was missing, and ChatGPT needed to fill it in). In only three cases, the question was not understood, which resulted in the lowest possible rating of 1. We noted that ChatGPT was very strong at recognizing the context. The notation of the output has always been the one given in the prompt.

MATH On the questions related to Algebra and Probability theory, ChatGPT got the reasoning often correctly. However, the most common type of error was **e4**: ChatGPT may struggle when confronted with standard operations, such as inverting fractions, least common multiples, changing the sign of numbers when moving them from one side of the equal sign to the other. Often, in these questions, a correct solution requires performing multiple operations in sequence. In most cases, at least one operation was wrong, preventing the model to get a rating of 5 on the output.

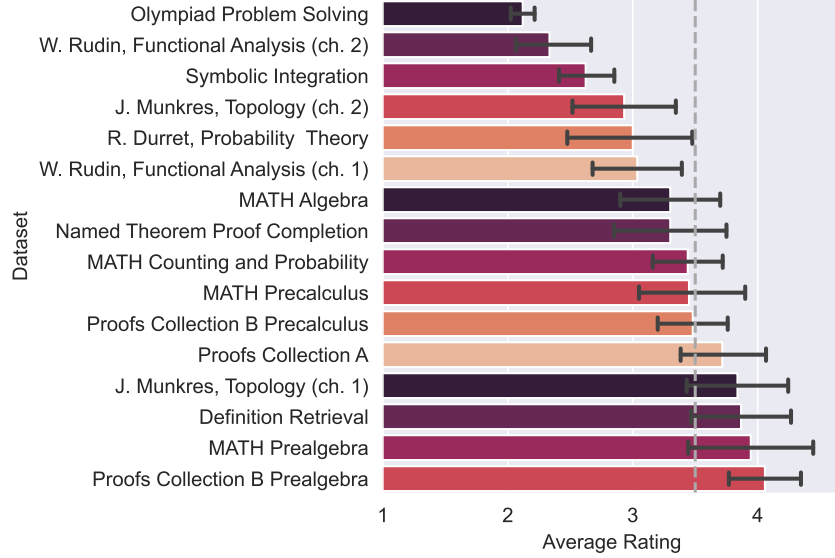


Figure 2: Average rating for each file in each subdataset. Since the maximal ranking is 5, and the minimal ranking where the question was at least understood is 2, a passing grade (50% of points) would be 3.5, as indicated by the dotted line.

Symbolic-Integration ChatGPT was dominated by systems that were trained specifically to solve integration problems [16]. In a number of instances ChatGPT got the structure of terms right (for example, the number of summands in the output, as well as where factors had to be placed before summands), but it failed at concrete computations. Even very simple examples were not correct. For example, the integral of $\frac{x^2}{2}$ is evaluated to $\frac{x^3}{3} + C$, where C is a constant of integration (the correct answer is $\frac{x^3}{6} + C$).

Named Theorem Proof Completion In this part of the *Search-Engine-Aspects* subdataset, ChatGPT knew almost all the theorems that it was asked at a basic level, but made mistakes when stating them. When it came to the proofs, it typically used way more than the necessary theory for the proofs, occasionally even results that only follow from the theorem which was asked for (error code `e5_5`).

Definition Retrieval In this part of the *Search-Engine-Aspects* subdataset, ChatGPT had a quite good performance: it recited most definitions correctly. It sometimes got confused when one asked about “distributions”, because it strongly favoured talking about distributions in the stochastic sense and not in the sense of the dual of test functions, as is common in functional analysis. Similarly for the adjective “closed”, where it chose to pick the context of algebra (instead of more topology) and interpreted it to mean “algebraically closed”.

4.1 Overall Performance

If we take a rating of 3.5 to be the threshold between success and failure, then Figure 2 shows that for most (in particular, harder) problems, ChatGPT will not pass. Two noteworthy exceptions are the *Definition Retrieval* file from the *Search-Engine-Aspects*, and the *J. Munkres, Topology (Ch. 1)* file from the *Grad-Text*, which comprise more advanced mathematics than the files from the other subdatasets that are above the 3.5 threshold. In particular, on problems that are within the style of mathematical olympiads, ChatGPT performs badly.

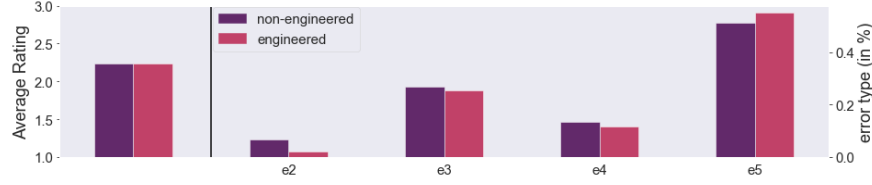


Figure 3: Effect of prompt engineering on the rating. Prompt engineering seems to only affect smaller errors that are not rooted in faulty logic.

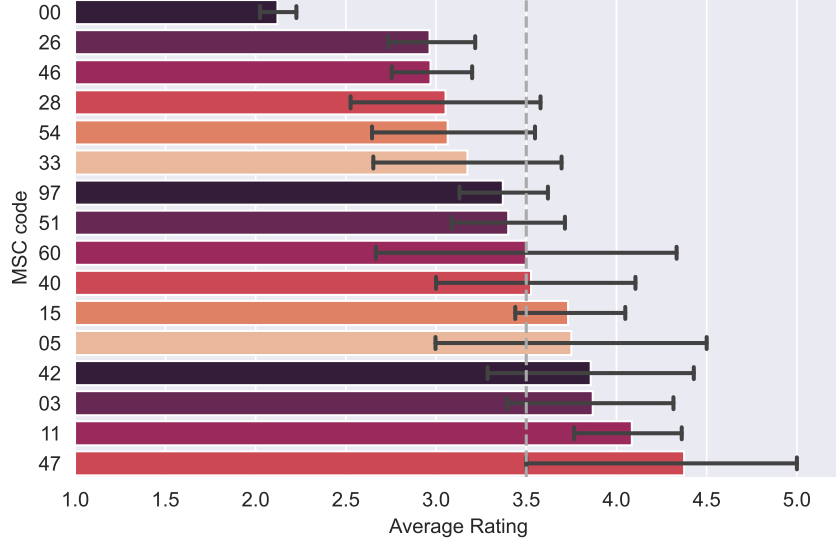


Figure 4: Average rating over mathematical fields (only those MSC codes that have at least 6 occurrences are shown).

The prompt length has no clear effect on the rating; see Figure 5. This indicates that ChatGPT’s rating is indeed reflective of mathematical difficulty.

Comparing ChatGPT to the performance obtained by [16], who obtained a percentage of correctly solved integral that was close to 100% on a collection of 500 test equations (table 3 from [16]), our random sample of their dataset (after translation from Polish notation and conversion to \LaTeX) achieves an average rating of 2.6, where a rating of 2 is dominating 60% of the time followed by a rating of 4 almost 20% of the time. These scores trail behind the performance that the model trained in [16] achieved.

The situation is similar when comparing ChatGPT to Minerva [17]. Their best model achieved an accuracy of 50% on the MATH dataset [14], table 3. On our random sample of the MATH dataset, ChatGPT achieves a perfect score only in 26% of cases, which implies that for us accuracy is 26%.

One interesting finding is related to prompt engineering. On 20% of the questions from *Olympiad-Problem-Solving* another prompt was added, where ChatGPT was prompted to proceed step-by-step. This had an interesting effect: Less severe errors **e2** went down, though more severe errors (**e5**-type errors) stayed about the same, as did the average rating, see Figure 3.

5 Conclusion

We have examined the behavior of ChatGPT across various datasets that test various aspect of mathematical skill. Contrary to the media sensation that ChatGPT has caused (see the Twitter counts from Figure 1),

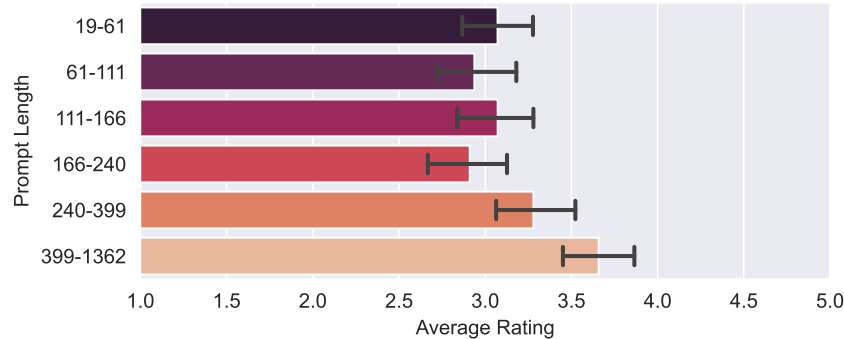


Figure 5: Prompt length (number of characters) has no significant effect on the rating.

ChatGPT is not yet ready to deliver high quality proofs consistently. At the same time, the quality of the answer can be positively surprising. In Section D from the appendix, we collect the best and the worst results for a number of selected datasets. The best responses can be seen to justify the media sensation. It seems fair to say that ChatGPT is *inconsistently* bad at advanced mathematics: While its ratings drop with the mathematical difficulty of a prompt, it does give surprising answers in a few cases.

However, ChatGPT falls short of achieving the same performance as models that are specifically trained for one single task. These models, in contrast, lack the flexibility of ChatGPT, that is a *universal* tool suitable for any area of mathematics.

We hope that the dataset that we release with this publication will spur on other professional mathematicians to contribute, in order to establish a thorough benchmark for assessing the mathematical abilities of LLMs. We will allow pull requests on our Github repository and encourage public participation.

References

- [1] Malik Amir, Yang-Hui He, Kyu-Hwan Lee, Thomas Oliver, and Eldar Sultanow. Machine Learning Class Numbers of Real Quadratic Fields. *arXiv preprint arXiv:2209.09283*, 2022.
- [2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020.
- [3] Francois Charton, Amaury Hayat, and Guillaume Lample. Learning advanced mathematical computations from examples. In *International Conference on Learning Representations*, 2021.
- [4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.
- [5] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

- [6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [7] Alex Davies, Petar Veličković, Lars Buesing, Sam Blackwell, Daniel Zheng, Nenad Tomašev, Richard Tanburn, Peter Battaglia, Charles Blundell, András Juhász, et al. Advancing mathematics by guiding human intuition with AI. *Nature*, 600(7887):70–74, 2021.
- [8] Jörg Denzinger, Matthias Fuchs, Christoph Goller, and Stephan Schulz. Learning from previous proof experience: A survey. Technical report, TU München, 1999.
- [9] Rick Durrett. *Probability: Theory and Examples*. 2019.
- [10] Arthur Engel. *Problem-Solving Strategies*. Springer, New York, 1998.
- [11] Timothy W. Gowers. It’s amusing when ChatGPT makes ridiculous mathematical mistakes. But of course, it’s more interesting to find out what it can do well. Here’s one example that wasn’t bad: I gave it a very rough outline of a proof and asked it to fill in the details., 2023.
- [12] John Harrison, Josef Urban, and Freek Wiedijk. History of interactive theorem proving. In *Computational Logic*, volume 9, pages 135–214, 2014.
- [13] Yang-Hui He. Machine-learning the string landscape. *Physics Letters B*, 774:564–568, 2017.
- [14] 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, 2021.
- [15] Tiffany H. Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, and Victor Tseng. Performance of ChatGPT on USMLE: Potential for AI-Assisted Medical Education Using Large Language Models. *medRxiv*, 2022.
- [16] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. *CoRR*, abs/1912.01412, 2019.
- [17] Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh 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. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [18] 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, July 2017. Association for Computational Linguistics.
- [19] Sascha Lobo. Das Ende von Google, wie wir es kannten, 2023.
- [20] Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. A survey of deep learning for mathematical reasoning. *arXiv preprint arXiv:2212.10535*, 2022.
- [21] Tranquil Sea Of Math. Does ChatGPT code LaTeX and write proofs? , 2022.
- [22] James R. Munkres. *Topology*. Prentice-Hall, Upper Saddle River, NJ, 2000.
- [23] Natalie. ChatGPT – Release Notes. <https://help.openai.com/en/articles/6825453-chatgpt-release-notes>, 2023. Accessed: 2023-01-22.

- [24] John Naughton. The ChatGPT bot is causing panic now – but it’ll soon be as mundane a tool as Excel, 2023.
- [25] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.
- [26] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online, June 2021. Association for Computational Linguistics.
- [27] Piotr Piękos, Mateusz Malinowski, and Henryk Michalewski. Measuring and improving BERT’s mathematical abilities by predicting the order of reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 383–394, Online, August 2021. Association for Computational Linguistics.
- [28] PowerfulJRE. What ChatGPT Could Mean for the Future of Artificial Intelligence, 2023.
- [29] Markus N. Rabe, Dennis Lee, Kshitij Bansal, and Christian Szegedy. Language modeling for formal mathematics. *CoRR*, abs/2006.04757, 2020.
- [30] Richard Van Noorden @richvn@mastodon.social. Huh. ChatGPT confidently gives the right kind of reasoning to solve this math problem, but whiffs on the algebra in the middle and gets the answer wrong., 2022.
- [31] Kevon Roose. The Brilliance and Weirdness of ChatGPT, 2022.
- [32] David Rozado. What is the IQ of ChatGPT? <https://davidrozado.substack.com/p/what-is-the-iq-of-chatgpt>, 2023. Accessed: 2023-01-23.
- [33] Walter Rudin. *Functional analysis*. McGraw-Hill, Inc, New York, 1991.
- [34] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.
- [35] John Schulman and Barret et al Zoph. ChatGPT: Optimizing Language Models for Dialogue, 2022.
- [36] Sarah Wiegrefe (sigmoid.social/@sarah). If text-davinci-001 is a rough approximate to the model reported in the NeurIPS 2020 paper, and text-davinci-002 is InstructGPT in the 2022 preprint, then what is just "davinci"? Trying to reproduce results from a time before this naming existed., 2020.
- [37] teddy. I made chatgpt take a full sat test. here’s how it did:, 2023.
- [38] Christian Terwiesch. Would Chat GPT3 Get a Wharton MBA? A Prediction Based on Its Performance in the Operations Management Course, 2023.
- [39] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.
- [40] Carroll Wainwright and Ryan Lowe. <https://github.com/openai/following-instructions-human-feedback>. <https://github.com/openai/following-instructions-human-feedback>, 2022.

Appendix

A Rating policy

Below is the policy that was followed by each assessor of ChatGPT’s output regarding the rating score:

- 0 → ChatGPT cannot process the query (due to consistently occurring timeouts or errors);
- 1 → failure to understand the query (“you ask it something about number theory and it responds with information about differential equations”);
- 2 → query was understood but answer was entirely wrong (“you ask what the prime divisors of 111 are⁶ and it responds with 8 and 6”);
- 3 → query was understood but answer was only partially correct (“you ask it what the prime divisors of 111 are⁶ and it responds with 3 and 6”);
- 4 → query was understood and answer was mostly correct (“you ask it what the prime divisors of 222 are⁷ and it responds with 3 and 37”);
- 5 → query was understood and answer was completely correct.

B Error and warning code policy

Below is the policy that was followed by each assessor of ChatGPT’s output regarding the error codes and warning codes:

Error codes:

- **e1** → missing examples (“you ask it what the prime divisors of 111 are⁶ and it responds with 3, missing 37”);
- **e2** → a few wrong statements (e.g., you ask it what the prime divisors of 30030 are⁸ and it responds with 2, 3, 5, 7, 13);
- **e3** → a lot of wrong statements (“you ask it what the prime divisors of 30030 are⁸ and it responds with 2, 5, 8, 12, 13, 15”);
- **e4** → wrong computations (an additional error flag to disambiguate between statements that are of computational nature or not);
- **e5** → denotes wrong logic or wrong flow of arguments, which we further subdivide into specific flags, as we prohibit the use of **e5** on its own, as it would be uninformative:
 - **e5_1** → ChatGPT claims that to complete a proof, statements need to be shown that are unrelated to the claim;
 - **e5_2** → a proof step is missing;
 - **e5_3** → an edge case has not been considered ;
 - **e5_4** → an inference step is not supported (e.g., ChatGPT claims that from A follows B, but this claim is not true);
 - **e5_5** → circular logical argument (using the hypothesis to prove the hypothesis);

⁶They are 37 and 3.

⁷They are 2, 37 and 3.

⁸They are 2, 3, 5, 7, 11.

- **e6** \rightarrow the general set-up is understood but the legal operations are not respected (e.g. we are given a puzzle where we're only allowed to add even integers but it change the rules and motivates the solution by allowing addition of odd integers).

Warning codes:

- **w1** → ChatGPT is withholding essential information related the prompt (“you asked it something about the integral $\int_{-\infty}^{\infty} e^{-x^2} dx$ and it answers correctly, but doesn’t tell you thing was actually a famous, named integral, the Gaussian integral)
- **w2** → ChatGPT is rambling (after answering (correctly or incorrectly) ChatGPT tells you much more than you wanted to know)
- **w3** → ChatGPT is hallucinating (after answering (correctly or incorrectly) ChatGPT tells you unrelated stuff)
- **w4** → weirdness (ChatGPT is being weird, for example adopting a strange tone of conversation or making opinionated statements)
- **w5** → it takes a number of tries to get ChatGPT to answer a prompt (because occasional timeouts or errors occur that are related to this particular prompt)
- **w6** → ChatGPT changes the notation from the prompt without being instructed to do so (e.g., the prompt contains a vector space \mathbf{X} , but ChatGPT calls it \mathbf{F})

C Further figures of ChatGPT’s performance

In this section, we collect further figures, which are not directly related to our discussion in the main body of the article, but which provide other views on the data, and other descriptive statistics.

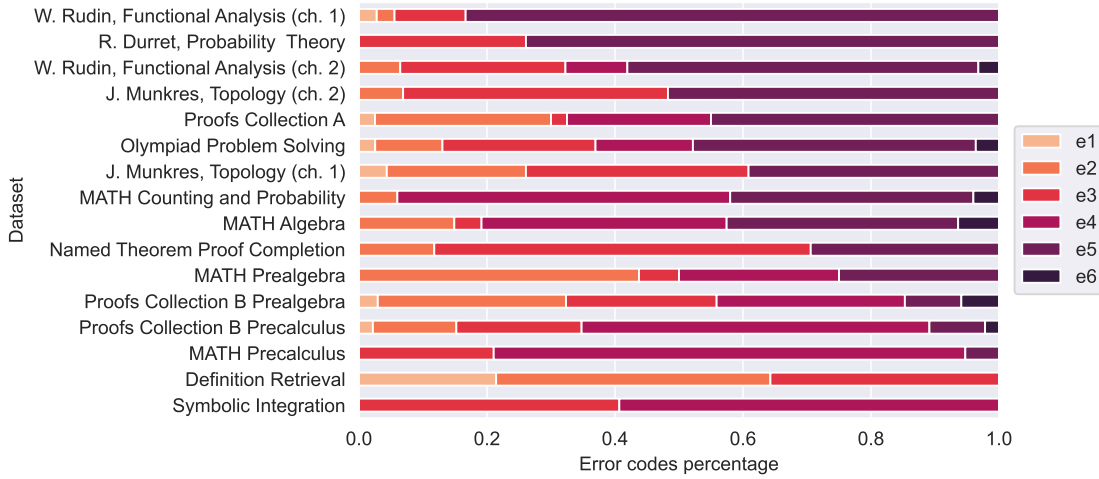


Figure 6: Error types per dataset sorted by frequency of "e5" (the most high-level error type).

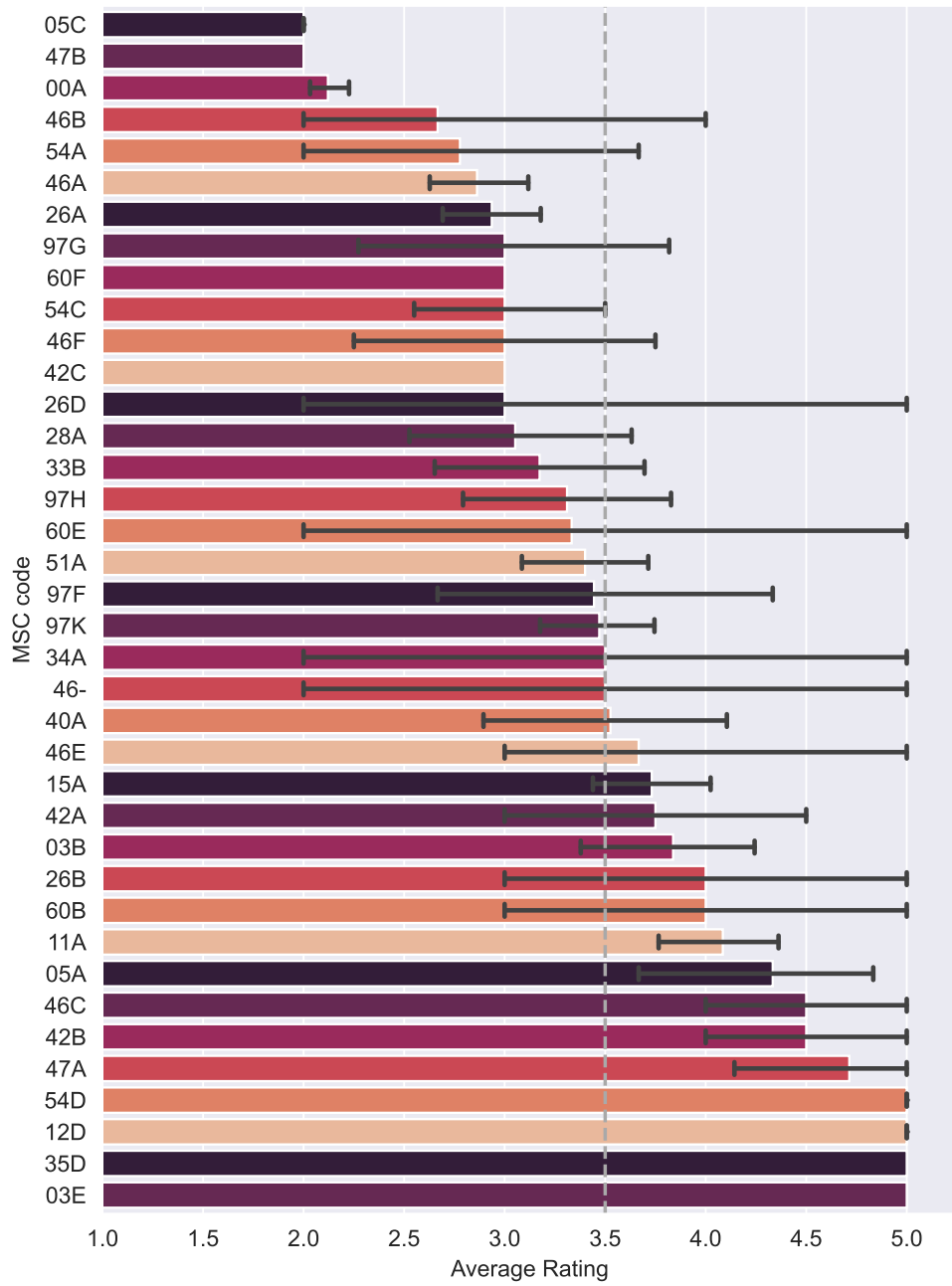


Figure 7: Rating by MSC codes (short version).

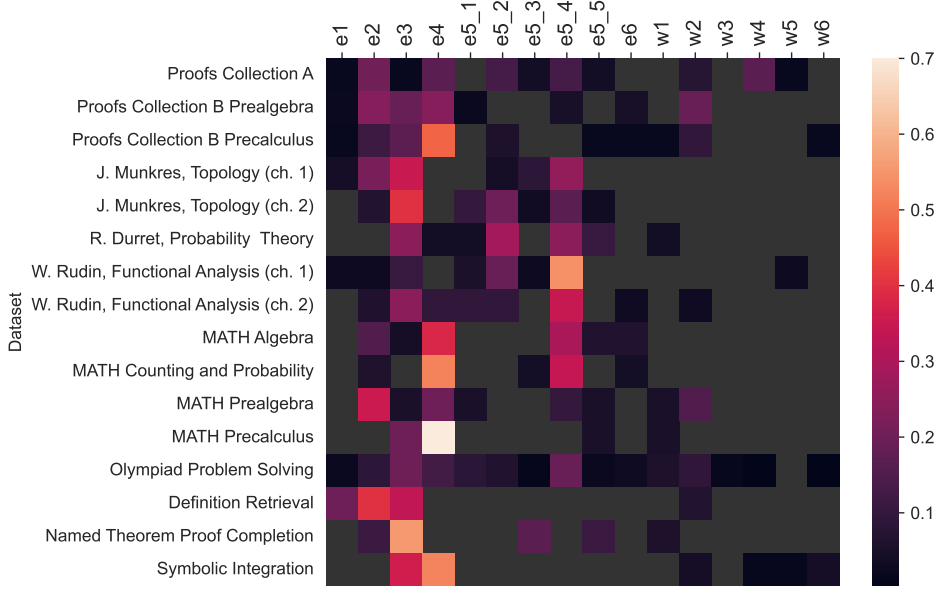


Figure 8: Error code percentages by subdatasets and files.

D Best-3 and worst-3 across selected subdatasets and files

From *Grad-Text* subdataset, comprising the books [33; 22; 9], best-3:

- "prompt": "Let μ be a measure, $\|f\|_{\infty} = \inf\{M : \mu(\{x : |f(x)| > M\}) = 0\}$ and $\|g\|_1 = \int |g| d\mu$. Prove that $\int |fg| d\mu \leq \|g\|_1 \|f\|_{\infty}$."

"output": "Let $M = \|f\|_{\infty}$ and $\epsilon > 0$. Then, by the definition of $\|f\|_{\infty}$, we have $\mu(\{x : |f(x)| > M + \epsilon\}) = 0$. Therefore, $\int |fg| d\mu = \int_{|f| \leq M + \epsilon} |fg| d\mu + \int_{|f| > M + \epsilon} |fg| d\mu \leq (M + \epsilon) \int |g| d\mu + \int_{|f| > M + \epsilon} |f| |g| d\mu$. The second term on the right side is zero because $\mu(\{x : |f(x)| > M + \epsilon\}) = 0$. So we have $\int |fg| d\mu \leq (M + \epsilon) \int |g| d\mu = (M + \epsilon) \|g\|_1$. As ϵ can be arbitrarily small, we can let it go to zero and we get $\int |fg| d\mu \leq M \|g\|_1 = \|f\|_{\infty} \|g\|_1$. Hence proved."

- "prompt": "If $\{T_{\alpha}\}$ is a family of topologies on X , show that $\bigcap_{\alpha} T_{\alpha}$ is a topology on X ."

"output": "To show that $\bigcap_{\alpha} T_{\alpha}$ is a topology on X , we need to show that it satisfies the three axioms of a topology: \emptyset and X are both in $\bigcap_{\alpha} T_{\alpha}$ since \emptyset and X are in every topology T_{α} . If $\{U_i\}_{i \in I}$ is a collection of sets in $\bigcap_{\alpha} T_{\alpha}$, then $\bigcup_{i \in I} U_i$ is also in $\bigcap_{\alpha} T_{\alpha}$. Since each T_{α} is a topology, $\bigcup_{i \in I} U_i$ is in every T_{α} , and so it is also in $\bigcap_{\alpha} T_{\alpha}$. If $\{U_i\}_{i \in I}$ is a collection of sets in $\bigcap_{\alpha} T_{\alpha}$, then $\bigcap_{i \in I} U_i$ is also in $\bigcap_{\alpha} T_{\alpha}$. Since each T_{α} is a topology, $\bigcap_{i \in I} U_i$ is in every T_{α} , and so it is also in $\bigcap_{\alpha} T_{\alpha}$. Therefore, $\bigcap_{\alpha} T_{\alpha}$ is a topology on X ."

- "prompt": "Let X be a vector space and A and B subsets of X . If A and B are balanced, so is $A + B$."

"output": "A subset S of a vector space X is called balanced if for every vector x in X and every scalar λ with $|\lambda| \leq 1$, λx is in S if x is in S .
To prove that $A + B$ is balanced, we must show that for every vector $x = a + b$ in $A + B$, and every scalar λ with $|\lambda| \leq 1$, $\lambda x = \lambda(a + b) = \lambda a + \lambda b$ is in $A + B$ if x is in $A + B$.
Since A and B are balanced, we know that for every a in A and every scalar λ with $|\lambda| \leq 1$, λa is in A if a is in A . Similarly, for every b in B and every scalar λ with $|\lambda| \leq 1$, λb is in B if b is in B .
So, if $x = a + b$ is in $A + B$, then a is in A and b is in B . Therefore, λa is in A and λb is in B , and thus $\lambda x = \lambda a + \lambda b$ is in $A + B$. Therefore, $A + B$ is balanced."

From *Grad-Text* subdataset, worst-3:

- "prompt": "Find a function $f : \mathbb{R} \rightarrow \mathbb{R}$ that is continuous at precisely one point."

"output": "One example of such a function is $f(x) = \begin{cases} x^2 & \text{if } x \neq 0 \\ 1 & \text{if } x = 0 \end{cases}$. This function is continuous at $x=0$ but not at any other point."

- "prompt": "Let X be a topological vector space. All sets mentioned below are understood to be the subsets of X . Prove the following statement: If A is compact and B is closed, then $A + B$ is closed"

"output": "A topological vector space is a vector space with a topology such that the vector addition and scalar multiplication are continuous.
To prove that $A + B$ is closed, we must show that the complement of $A + B$ is open.
Let $y \in X \setminus (A + B)$. Then there exists some x_1 in A and x_2 in B such that $y = x_1 + x_2$. Since A is compact and B is closed, there exist neighborhoods U_1 of x_1 and U_2 of x_2 such that U_1 is relatively compact and $X \setminus U_2$ is open.
Now consider the set $V = U_1 + U_2 = \{x_1 + x_2 : x_1 \in U_1, x_2 \in U_2\}$. This set is an open neighborhood of y and it is contained in $X \setminus (A + B)$, which shows that $X \setminus (A + B)$ is open and hence $A + B$ is closed."

- "prompt": "Let X be a topological vector space. All sets mentioned below are understood to be the subsets of X . Prove the following statement: The sum of two closed sets may fail to be closed"

"output": "A counterexample can be used to prove that the sum of two closed sets may fail to be closed.
For example, let $X = \mathbb{R}$ and let $A = [0, 1]$ and $B = (1, 2]$. Both A and B are closed sets in \mathbb{R} , but $A + B = [0, 2]$ is not a closed set in \mathbb{R} ."

From the *Aperture-Proofs* subdataset, *Proofs Collection A* file, best-3:

- "prompt": "We want to show that any open subset of \mathbb{R} is a countable union of disjoint open intervals. Let U be a non-empty open subset of \mathbb{R} . For $x, y \in U$ define $x \sim y$ iff $[x, y] \subset U$. It's easily checked that \sim is an equivalence relation on U whose equivalence classes are pairwise disjoint open intervals in \mathbb{R} . (The

term *interval* here includes unbounded intervals, i.e., rays.) Let \mathcal{I} be the set of \sim -classes. Clearly $U = \bigcup_{I \in \mathcal{I}} I$. For each $I \in \mathcal{I}$ choose a rational $q_I \in I$; the map $\mathcal{I} \rightarrow \mathbb{Q} : I \mapsto q_I$ is injective, so \mathcal{I} is countable. Can you fill in the missing condition?"

"output": "The missing condition is \"there exists an open interval containing x and y that is contained in U \" "

- "prompt": "The p -norm on \mathbb{R}^n is given by $\|x\|_p = \left(\sum_{k=1}^n |x_k|^p \right)^{1/p}$. We want to show that $\|x\|_p \leq C \|x\|_q$ for arbitrary $1 \leq p < q$.
We compute $\|x\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \leq \left(\sum_{i=1}^n |x_i|^q \right)^{\frac{p}{q}} n^{1-\frac{p}{q}} = \|x\|_q^p n^{1-\frac{p}{q}} = \|x\|_q^{p-1} \|x\|_q$. Which result has been used in the inequality above?"

"output": "The inequality used in the above calculation is Hölder's inequality. It states that for any $p, q \geq 1$ such that $\frac{1}{p} + \frac{1}{q} = 1$, and for any vectors $x, y \in \mathbb{R}^n$, $\left| \sum_{i=1}^n x_i y_i \right| \leq \left(\sum_{i=1}^n |x_i|^p \right)^{1/p} \left(\sum_{i=1}^n |y_i|^q \right)^{1/q}$."

"rating": "5",

- "prompt": "Let $f_n(x) = \frac{x^2}{x^2 + (1-nx)^2}$ $\forall x \in [0, 1]$, $n=1, 2, 3, \dots$. $\lim_{n \rightarrow \infty} f_n(x) = 0$ $\forall x \in [0, 1]$, but there is a sequence $\{x_n\}$ with $f_n(x_n) = 1$ $\forall n=1, 2, 3, \dots$ so that no subsequence of $\{f_n\}$ can converge uniformly on $[0, 1]$. Can you define a suitable sequence $\{x_n\}$?"

"output": "One possible choice for the sequence $\{x_n\}$ is $x_n = \frac{1}{n}$. This can be verified by plugging in $x_n = \frac{1}{n}$ into the definition of $f_n(x)$ and seeing that $f_n(x_n) = 1$ for all n . Since this sequence does not have a limit, it can be shown that no subsequence of $\{f_n\}$ can converge uniformly on $[0, 1]$."

From the *Aperture-Proofs* subdataset, *Proofs Collection A* file, worst-3:

- "prompt": "We want to compute $\sum_{n=1}^{\infty} \frac{2n}{3^{n+1}}$.
Let's first find a general formula for the following sum: $S_m = \sum_{n=1}^m nr^n$.
Notice that $nS_m - rS_m = -mr^{m+1} + \sum_{n=1}^m r^n$
& = $-mr^{m+1} + \frac{r-r^{m+1}}{1-r}$
& = $\frac{mr^{m+2} - (m+1)r^{m+1} + r}{1-r}$.
Hence $S_m = \frac{mr^{m+2} - (m+1)r^{m+1} + r}{(1-r)^2}$. Can you finish the calculation?"

"output": "Yes, to finish the calculation we can substitute the values of $m = \infty$, $r = \frac{2}{3}$ into the formula we found for S_m : $S_{\infty} = \frac{\infty \left(\frac{2}{3} \right)^{\infty+1} - (\infty+1) \left(\frac{2}{3} \right)^{\infty} + \frac{2}{3}}{\left(1 - \frac{2}{3} \right)^2}$. This simplifies to $S_{\infty} = \frac{\frac{2}{9}}{\left(\frac{1}{3} \right)^2} = 2$."

- "prompt": "Suppose V is an inner product space and $T \in \mathcal{L}(V)$. We want to prove that $|\det(T)| = \det(\sqrt{T^* T})$. By the Polar Decomposition there is an isometry $S \in \mathcal{L}(V)$ such that $T = S \sqrt{T^* T}$. Can you finish the proof?"

"output": "Yes, once we have that $T = S \sqrt{T^* T}$, we can take the determinant of both sides to get $\det(T) = \det(S) \det(\sqrt{T^* T})$. Since S is an isometry, it preserves inner products, so $\det(S) = \pm 1$. Therefore, $\det(T) = \pm \det(\sqrt{T^* T})$. Since the determinant of a matrix is always non-negative, we have that $|\det(T)| = \det(\sqrt{T^* T})$ "

- "prompt": " F is a finite field of order q . We want to compute the size of $\operatorname{GL}_n(F)$. In order for an $n \times n$ matrix to be invertible, we need the rows to be linearly independent. As you note, we have $q^n - 1$ choices for the first row; now, there are $q^n - q$ vectors in the span of the first row, so we have $q^n - q$ choices for the second row. Now, let v_1, v_2 be the first two rows. Then the set of vectors in the span of v_1, v_2 is of the form $\{c_1 v_1 + c_2 v_2 \mid c_1, c_2 \in F\}$. This set is of size q^2 , as we have q choices for c_1 and q choices for c_2 . Thus, we have $q^n - q^2$ choices for the third row. Continuing this gives the desired formula. Can you state the formula?"

"output": "The size of $\operatorname{GL}_n(F)$ is given by the formula $(q^n - 1)(q^{n-1} - q)(q^{n-2} - q^2) \cdots (q^2 - q)(q - 1)$."