MATPROVE Dataset - Mathematical Problem-solving Dataset of Lessons and Exercises

Anonymous review submission

Anonymous review submission

Abstract

MATPROVE is a dataset featuring 57 lessons and 5221 exercises in topics such as tensor calculus, discrete mathematics, or linear algebra, aimed at training and evaluating machine learning techniques in mathematical problem solving and theorem proving. This dataset provides an ideal training ground for models to learn and apply theorem proving techniques as described in human-readable form. GPT-4 solves 68.9% of these exercises when shown the topic lesson, and 61.4% unaided.

1 Introduction

Advances in Deep Learning, significantly propelled by large datasets such as ImageNet [9], Netflix's movie recommendation [4], and the Stanford Sentiment Treebank [21], have paralleled developments in Theorem Proving with resources now available for enhancing Large Language Models (LLMs) to compete at High School mathematics [24, 3, 23]. In order to benefit advanced mathematics, databases for research-level proofs such as Lean [8] and Coq [5] are expanding, but a widening gap exists between human-readable MWP (Math Word Problems) and machine-readable TP (Theorem Proving). MATPROVE aims to bridge this by providing a structured corpus of university-level mathematics for model training in both unsupervised and supervised contexts.

2 Prior Work

Significant strides in automated theorem proving and mathematical reasoning have been facilitated by specialized datasets like ImageNet in other fields [3]. In mathematics, datasets such as GSM8K, MathQA, and ASDiv-A have revolutionized LLMs' capabilities [7, 2, 16]. Moreover, the integration of informal language to guide formal proofs and understanding diagrams in mathematics underscore the necessity for interdisciplinary approaches [12, 14, 18, 17, 13]. Recent workshops and surveys like the MATH-AI series and mathematical reasoning workshops have highlighted ongoing research, facilitating collaboration and setting new directions. Additionally, benchmarks such as Math23K and Dolphin18K are proving crucial in advancing NLP models to tackle diverse mathematical problems, ensuring robust and versatile AI models [22, 20].

3 Method

First, twelve textbooks in undergraduate and graduate mathematics were selected from the university library, such that they contain a comprehensive review of their subject, as well as a large collection of worked problems. Following a thorough review of approaches to convert pdf files into latex [15, 11, 10, 6], the commercial mathpix service was selected. Each book was processed and converted into a single .tex file and many image files for figures.

Next, the latex files were manually reviewed, and 5 were dropped for containing minor errors. Then, a python regular expression processor was written for each book to split it into chapters, and split each chapter into lesson, solved problems, and other. Some books contain solutions to problems at the end, and were processed accordingly. Three of the remaining seven books were not easily processed with regular expressions and were dropped.

Finally, each solved problem was processed with GPT-4 [1], prompted to split it into question, work, and answer. Some problems are composed of multiple sub-problems (ex.: a), b), c)...), and these were not separated because they often refer to each other.

The code is provided for review of implementation details, and to enable the creation of more data.

4 Dataset

Comprising material from four textbooks, the dataset includes 57 lessons and 3302 problem-answer pairs, 1642 of which contain fully worked solutions. Overall, the dataset encapsulates 5221 gradable question-answer pairs, enriched with figures for certain problems, providing a substantial resource for developing AI capable of complex mathematical reasoning.

Below is a randomly selected problem from the dataset from a textbook [19]. Figure 1 shows the raw LaTeX code of the problem, figures 2 and 3 show how the problem is split into question and answer, in rendered LaTeX.

```
14.8. Prove Theorem 14.1: Suppose $S$ is a finite poset with $n$ elements.
Then there exists a consistent enumeration $f: S \rightarrow\{1,2, \ldots, n\}$.
```

Figure 1: LaTeX code of the example problem question segment

Prove Theorem 14.1: Suppose S is a finite poset with n elements. Then there exists a consistent enumeration $f : S \to \{1, 2, \ldots, n\}$.

Figure 2: Compiled LaTeX of example problem question segment

The proof is by induction on the number n of elements in S. Suppose $n = 1$, say $S = \{s\}$. Then $f(s) = 1$ is a consistent enumeration of S. Now suppose $n > 1$ and the theorem holds for posets with fewer than n elements. Let $a \in S$ be a minimal element. (Such an element a exists since S is finite.) Let $T = S\backslash\{a\}$. Then T is a finite poset with $n - 1$ elements and hence, by induction, T admits a consistent enumeration; say $g : T \to \{1, 2, \ldots, n - 1\}$. Define $f : S \to \{1, 2, \ldots, n\}$ by:

$$f(x) = \begin{cases} 1, & \text{if } x = a \\ g(x) + 1 & \text{if } x \neq a \end{cases}$$

Figure 3: Compiled LaTeX of example problem solution segment

5 Evaluation

A consequence of the real-world nature of the MATPROVE dataset is that answers come in a wide variety of formats:

1. Yes/No
2. Multiple choice
3. Numerical (i.e., a single number or vector)
4. Text
5. Drawing
6. Proof

Therefore, the proposed evaluation of a particular set of answers is by using a particular LLM to compare the proposed answers with the reference answers in the dataset. The technique implemented in the test code uses GPT-3.5 to grade each of the 5221 question-answer pairs as follows: 0-no answer or wrong, 1-work correct but answer wrong, 2-correct answer.

6 Baseline

GPT-4 is known to give state-of-the-art results in mathematical problem solving and reasoning, so it was used here to provide a baseline. GPT-4 was prompted to show work and provide the answer to each question, and the answers were then graded as described above. The same was performed when providing GPT-4 with the lesson, which resulted in a 12% relative improvement.

Condition	**Correct Answers**	**Accuracy**
Without Lesson	3210	61.4%
With Lesson	3595	68.9%

Table 1: Performance of GPT-4 With and Without Initial Lesson

The generation code, test code, raw data and formatted dataset are all available at (github link removed for anonymous review) with usage instructions.

7 Future Work and Conclusion

In order to advance Artificial Intelligence for Theorem Proving, new models must be devised to take advantage of the corpus of mathematics textbooks. In order to design and train such models, a dataset of lessons and relevant solved problems is necessary. Through manual selection, data-mining techniques, and prompting GPT-4, such a dataset has been created for a corpus of 57 lessons in varied university-level topics in mathematics, supplemented by 5221 unique problem-answer pairs.

Future work will include extending the dataset by repeating the process outlined here, gradually producing a dataset containing cutting-edge mathematical techniques and their example applications. This will bridge the gap between human-readable and machine-readable mathematics, enabling the development of techniques to autonomously learn, make use of, and develop new theorems and proofs in human-readable format.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

[3] BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*, 2023.

[4] James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, 2007.

[5] Yves Bertot and Pierre Castéran. *Interactive theorem proving and program development: Coq'Art: the calculus of inductive constructions*. Springer Science & Business Media, 2013.

[6] Lukas Blecher. pix2tex: Latex-ocr using deep learning, 2024. Accessed: May 10, 2024.

[7] 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.

[8] Leonardo De Moura, Soonho Kong, Jeremy Avigad, Floris Van Doorn, and Jakob von Raumer. The lean theorem prover (system description). In *Automated Deduction-CADE-25: 25th International Conference on Automated Deduction, Berlin, Germany, August 1-7, 2015, Proceedings 25*, pages 378–388. Springer, 2015.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Yuntian Deng, Anssi Kanervisto, Jeffrey Ling, and Alexander M. Rush. Image-to-markup generation with coarse-to-fine attention, 2017.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[12] Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*, 2022.

[13] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

[14] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

[15] Mathpix. Mathpix: Transform images and pdfs into latex, 2024. Accessed: May 10, 2024.

[16] Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*, 2021.

[17] Mrinmaya Sachan, Kumar Dubey, and Eric Xing. From textbooks to knowledge: A case study in harvesting axiomatic knowledge from textbooks to solve geometry problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 773–784, 2017.

[18] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1466–1476, 2015.

[19] Lipschutz Seymour and Lipson Marc Lars. *Theory and problems of discrete mathematics*. Schaums Outline Series Mcgraw Hill, 2007.

[20] Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1132–1142, 2015.

[21] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

[22] Yan Wang, Xiaojiang Liu, and Shuming Shi. Deep neural solver for math word problems. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 845–854, 2017.

[23] Kaiyu Yang and Jia Deng. Learning to prove theorems via interacting with proof assistants. In *International Conference on Machine Learning*, pages 6984–6994. PMLR, 2019.

[24] Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. The gap of semantic parsing: A survey on automatic math word problem solvers. *IEEE transactions on pattern analysis and machine intelligence*, 42(9):2287–2305, 2019.