Dean&Francis
ISSN 2959-6157

Challenges and Improvements in Mathematical Reasoning for Large Models

Ziming Wang

School of Software, Xinjiang University, Urumqi, Xinjiang, China
*Corresponding author: 20232204202@stu.xju.edu.cn

Abstract:

Mathematical reasoning is a fundamental criterion for AI, and this is also a problem to be solved before arriving at AGI. Large Language Models(LLMs) have become prominent in this domain.. LLM's benchmark performance (GSM8K, MATH): However, a large number of investigation indicate that they only mimic what the training data are, so they do not properly understand logical problems, and that means their reasoning ability is extraordinarily flaky, meaning that even a small perturbation in how problems are phrased could completely alter the reasoning which is really the difference between being smart statistically and understanding. So, if you are going to try and reason around these sorts of limitations, to find a way to a more robust application of math intelligence, you need to consider all research directions being explored right now. Analysis is broken down into 3 directions:(1) The first direction involves methods that include tweaking methods from a processing intervention standpoint towards reasoning chains. (2). The second direction is placeholder self-verification and reflection, where the models are urged to think about and improve their own reasoning. (3). The third direction is scientific verification and a measure of efficiency, focusing on thorough benchmarking and ways to test like tool integration. This survey is meaningful, as it is on a topic that is ever-changing, and will provide a baseline to help a lay audience, in addition to showcasing what the state-of-the-art approaches are, while still critically examining whether it can overcome any of the inherent challenges associated with LLMs. I hope it can be useful in conducting research in the future for more logical/more trustworthy AIs.

Keywords: Mathematical reasoning, LLMs, Chain-of-Thought, self-verification, AI Benchmarking..

ISSN 2959-6157

1. Introduction

Mathematical Reasoning is an essential marker for Artificial Intelligence as well as a challenging task to reach AGI Large-Language-Model LLMs have emerged and greatly developed the field.While these models excel in pattern recognition and can solve many standardized problems, their approach often lacks true logical understanding. They primarily rely on statistical correlations learned from data, rather than structured reasoning processes, which leads to unstable performance and susceptibility to the way problems are phrased.

This limitation stems from the fundamental architecture and training method of large language models. Most large language models are based on the Transformer architecture, which essentially utilizes statistical information from large datasets to predict the next word [1]. This probability-based mechanism fundamentally differs from the deterministic reasoning required by mathematical thinking. Research has shown that such models lack explicit symbolic processing and logical deduction capabilities [2], and their reasoning process is closer to "System 1" (fast, automatic intuitive thinking) rather than "System 2" (slow, logical analytical thinking) required for mathematical problems [3]. Therefore, even if the model can imitate the linguistic form of mathematical proofs, it often lacks a true understanding of the concepts involved [4].

This fundamental limitation leads to frequent errors in practical applications of models, such as basic arithmetic errors, ignoring explicit constraints in the problem, or introducing unfounded assumptions to apply known formulas. To build more robust and reliable AI systems capable of reasoning, researchers are exploring solutions from multiple directions. This paper focuses on reviewing the following three key strategies: (1) process intervention and optimization (such as advanced prompting techniques, reasoning chain fine-tuning); (2) self-verification and self-reflection (self-checking and correction during the reasoning process); (3) scientific evaluation and efficient execution (such as strong benchmark integration, tool invocation). By examining these interrelated methods, this paper aims to clearly outline the current efforts to bridge the gap between statistical pattern matching and genuine mathematical reasoning.

In summary, the limitations of large language models in mathematical reasoning are deeply rooted in their statistical and pattern-matching nature. To effectively address these challenges, it is no longer sufficient to rely solely on expanding model size or increasing data volume. Instead, it is necessary to develop more hybrid, structured, and inherently verifiable reasoning frameworks - which also provides the research background and direction for the intervention and optimization strategies that will be elaborated on in the following sections.

2. Process Intervention and Optimization

To make LLMs "smarter" in mathematical reasoning - meaning achieving this via process intervention and optimization - it isn't enough to just have the models trained with the final answers; it requires deeper thinking into the "why" behind each step of reasoning. It is also based on the recognition that complex problems in mathematics are ultimately from a series of logical reasoning, and the final number arrived at could, in theory, be reasonable from flawed reasoning (i.e., "false correctness"). If people can intervene and resolve these degrees of incorrect inference, it will produce models that are better as well as more interpretable. It fundamentally gives feedback on the reasoning behind the model's thought process, in turn improving their ability to solve problems through more rewards for correct reasoning - instead of merely being "correct". This emerging space focuses on a new modeling area on producing high-quality reward models, which effectively playfully do as the name suggests, prompting the LLM or guiding it along the way, each step of the way, until a final solution generation.

A foundational technique in this area is using Process-based Reward Models (PRMs). Research conducted at OpenAI has introduced PRMs, whose purpose is to evaluate and score performance in each individual reasoning step in a model's logical chain of reasoning [7]. A PRM does not simply give a reward for just one final correct answer, but the PRM evaluates the accuracy of every step in the reasoning. For example, when a student is solving a multi-step algebra problem, a student might use a correct formula in the calculation of a step, only to make a mistake in the next step, leading to an overall correct answer. In this scenario, a PRM would still offer a reward to the student for using the correct formula, even though they made an arithmetic mistake. More granular levels of feedback are key to letting the model learn to reason correctly. The paper "Let's Verify Step By Step" demonstrated that training with a PRM was even more effective than training the model on outcomes alone, which produced greater performance and overall better alignment with human preferred reasoning methods[7].

ORM, or Outcome-Based Reward Model, is an alternative to the PRM, which focuses on the final outcome versus the individual reasoning process; nevertheless, it is typically used in combination with the PRM to create a model whose reasoning makes sense according to its reasoning

process. A nice mixed feedback helps supply the model with a stronger training signal. A large advance in this world is using RL with human feedback on the reasoning process: [8] In this world, humans read through and give rankings to all of the different thought chains that a model is coming up with and then use the ranking to train the reward model. After that, develop the reward model to fine-tune the LLM through a reinforcement learning method, such as PPO [8], to reinforce reasoning that is in concert with human reasoning. The main idea here is that people find it much easier to point out a single mistake in a long line of reasoning versus coming up with a perfect answer from scratch, which means that this is something that can scale to supervise other agents. These are key process driven moves to develop LLMs out of clever pattern recognizers and towards better and more trustworthy mathematical reasoners.

3. Self-Assessment and Reflection

The Self-Assessment and Reflection strategy develops an ability for an LLM to introspect, critique, and revise its erroneous mathematical reasoning. This framework mirrors the human process of checking the work after an answer is curated through double-checking. Instead of generating a solution in one shot, the model will be tasked with generating an earlier solution and then thinking rigorously about that answer, or, generating multiple alternative answers and choosing the best one. Internal feedback loops help with some common issues, like making mistakes when doing math, reasoning incorrectly, or not understanding what the problem is asking for. That capacity for reflection and assessment facilitates a higher level of accuracy and overall robustness without having to receive immediate feedback from the external, which makes the model more reliable and autonomous reasoning.

One of the best examples of this work is self-consistency. The concept is introduced in the paper "Self-Consistency Improves Chain-of-Thought Reasoning of Language Models" and consists of asking the model to produce multiple and different chains of thought to solve the same problem, and then to take the majority response as the final prediction [9]. The rationale is that when a model makes an error on a specific chain of reasoning, it likely will not make that same error on another chain of reasoning. The hope is to dampen those erroneous chains of reasoning by using a majority vote to determine final answers once the model generates answers from different chains of reasoning. This method does a very good job of improving effectiveness and accuracy for tests about numbers, everyday logic, and various types of special thinking [9].

After that, researchers began doing many, many more reflective prompts. For example, researchers can now explicitly prompt the models to act as the "verifier" of their own solutions. The model works to think up a first solution, and then is prompted, in the style of an evaluator: "look closely to find weaknesses in solutions" or "see if you can find any errors." This second (or third level) prompt makes the model re-examine the problem, and uniformly finds better approaches, and, often through re-evaluation, identifies errors that might have slipped through the first iteration. Another layered, or enhanced method is the creation of an "internal dialogue," where the model proposes a series of solution steps, a different part of those models (or special models) critiques those action options, and the original model proposes a solution that improves upon those evaluations. To further tighten it within a loop again and this can also be called (like a tree of thoughts) if it is like people are taking on (it may have lots of steps, which raises possibilities for mistakes to pile up) [10]. These self-correction ways are the first key steps toward building a more autonomous, dependable AI system for mathematics.

4. Scientific Evaluation and Effective Disentangling

If LLMs are to make real shifts in the field of mathematical reasoning and develop solutions that are scaled or practical and effective, then a rigorous scientific evaluation must be systematized. This means developing specific measures so that people can determine the level of performance of different models and methods to address real tasks. A real scientific evaluation must be different than the surface narratives of success and be on the road to developing quantifiable verification indicators: systems should be developed that not only reveal weak points but also treat the various models as a reciprocal organic extensions to real reasoning—knitting systems together via linked data verification. At the same time, as models are continuously expanding in size, and model costs (the generation of many routes for reasoning) remain ever higher, developing research on effective disentangling has become important—getting great performance without excessive or high computational cost (to apply more reasoning efficiencies into the real world).

Scientific evaluation in this field relies on benchmarks that have progressively increased in complexity. Previously, benchmarks only involved basic operations like addition, but it has gotten more complicated. An example of the benchmarks is the GSM8K, a common benchmark that consists of grade school-level math word problems requiring multi-step reasoning [11]. Being skilled in converting

ISSN 2959-6157

prose to math and numbers with word problems represents the benchmark tests of natural language. To take it a step further, there is also a MATH dataset called Measuring Mathematics Ability for Trainers, which consists of high school-level competition problems in Algebra, Geometry, Number Theory and Calculus [12]. This benchmarks a harder abstraction and more complicated reasoning to benchmark ability. MATH, in addition to assessing the arithmetic answer, is also attempting to assess if the model's intermediate steps, completed along with their final answers, are correct [7]. These benchmarkers attempt to prevent people from "benchmarked hacking," optimizing a model to perform better on a certain benchmark rather than performing at the highest level through generalization.

To the efficiency of the cost of human labor, it might be easier than the approach of complexity, like self-consistency, as self-consistency needs many resamplings [9]. People are testing things like speculative decoding [13,14] with distilled reasoning: Why train a smaller and much cheaper Student Model using a bigger and more powerful Teacher Model? Student Model so that it can use lesser computational cost while still learning to mimic the better reasoning performance of the teacher. Furthermore, the third major line of research is providing greater access to resources for models, namely allowing a model to use tools, like a Python interpreter, to do calculations for it, to protect against the inaccuracy or incorrectness of doing these calculations themselves [15, 5]. To increase the utility of empirical work, an interactive combination of purely evaluative and goal-oriented exercises is needed to move new research from the lab to these useful tools that can produce reliable outputs to solve math problems.

5. Conclusion

This study analyzes many new issues and solutions in the math reasoning of Large Language Models. The paper started with the base issue: LLMs typically are probabilistic matchers and therefore are often unaware of what is symbolized to make logical breakdowns. As a consequence, there are often gross errors, such as digital arithmetic performance that declines, and missing important elements from inputs.

The paper has addressed a few of these themes in 3 research directions. Firstly, Process Intervention and optimization-wise, it starts with a series of research indicating models have moved from outcome-based feedback in developing language models to process-based feedback. PRM and RLHF, and other such methods, are useful to demonstrate and develop meta-cognitive reasoning about the value of certain types of reasoning, which is then developed into better or clearer means of reasoning to solve problems on paper. The second research line is Self Verification & Self Refinement, which essentially enables the model to self-critique and self-revise its work and learning experience; and the model gets feedback to self-evaluate its correctness in its use of reasoning, such as self-consistency or prompting verification, as well as improving autonomy. Lastly, the last research line of Scientific Evaluation and Efficient Implementation reminds us that more rigor is required on datasets the scale of MATH and GSM8K, and more efficient inference processes, such as model distillation, are required to see these types of capabilities in practice.

These research dimensions present a full route map for continuing exploration. What people are not arguing for are separate and additive reasons that improve mathematical thinking capacity, but ultimately will require an ecosystem that embraces the small changes when people are not behaving as fully autonomous agents, that critiques the mistakes, and checks the outputs either way. In such a way, a heuristic route toward AI systems that can solve complex math problems because they can produce proven/logical, or true system outputs.'

References

[1] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, et al. Attention is all you need. Advances in Neural Information Processing Systems, 2017, 30.

[2] Kahneman D. Thinking, fast and slow. Macmillan, 2011.

[3] Bubeck S, Chandrasekaran V, Eldan R, Gehrke J, Horvitz E, Kamar E, et al. Sparks of artificial general intelligence: Early experiments with GPT-4. arXiv:2303.12712, 2023.

[4] Wei J, Wang X, Schuurmans D, Bosma M, Xia F, Chi E, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in Neural Information Processing Systems, 2022, 35: 24824–24837.

[5] Lightman H, Kosaraju V, Burda Y, Edwards H, Baker B, Lee T, et al. Let's verify step by step. The Twelfth International Conference on Learning Representations, May 2023.

[6] Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 2022, 35: 27730–27744.

[7] Wang X, Wei J, Schuurmans D, Le Q, Chi E, Narang S, et al. Self-consistency improves chain of thought reasoning in language models. arXiv:2203.11171, 2022.

[8] Yao S, Yu D, Zhao J, Shafran I, Griffiths T, Cao Y, Narasimhan K. Tree of thoughts: Deliberate problem solving with large language models. Advances in Neural Information Processing Systems, 2023, 36: 11809–11822.

[9] Cobbe K, Kosaraju V, Bavarian M, Chen M, Jun H, Kaiser

L, et al. Training verifiers to solve math word problems. arXiv:2110.14168, 2021.
[10] Hendrycks D, Burns C, Kadavath S, Arora A, Basart S, Tang E, et al. Measuring mathematical problem solving with the math dataset. arXiv:2103.03874, 2021.
[11] Zheng Z, Ning K, Wang Y, Zhang J, Zheng D, Ye M, Chen J. A survey of large language models for code: Evolution, benchmarking, and future trends. arXiv:2311.10372, 2023.
[12] Schick T, Dwivedi-Yu J, Dessì R, Raileanu R, Lomeli M, Hambro E, et al. Toolformer: Language models can teach themselves to use tools. Advances in Neural Information Processing Systems, 2023, 36: 68539–68551.
[13] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, et al. PAL: Program-aided language models. International Conference on Machine Learning, July 2023: 10764–10799.
[14] Zhou Z, Ning X, Hong K, Fu T, Xu J, Li S, et al. A survey on efficient inference for large language models. arXiv:2404.14294, 2024.
[15] Olausson T X, Gu A, Lipkin B, Zhang C E, Solar-Lezama A, Tenenbaum J B, Levy R. LINC: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. arXiv:2310.15164, 2023.