
LLM for Math Reasoning

- Large Language Models for Mathematical Reasoning: Progresses and Challenges
- DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

Large Language Models for Mathematical Reasoning: Progresses and Challenges

Contents

- Introduction
- Math Problems & Datasets
- Related Work
- Methodologies
- Analysis
- Challenges
- Conclusion

Chenxu Li (jnr2jp)

Introduction

Background



$\pi = (r-e)^2$ $\sqrt{\pi}$ $f(x) = ax^2 + bx + c$ r $(x+y)^2 - (x-y)$ $E = mc^2$

$\int \beta \sqrt{9} \quad y = 3x+6$ $\frac{a}{\sqrt{b}}$ $\frac{1}{5} + \frac{3}{6}$ y a

y $\frac{2}{3} + \frac{4}{6}$ **MATH** x b

$x+y=3$ $Y = x^2 - \frac{384}{153}$ ψ

$Y = \cos x - \sin x$ $Me = x + B \left[\frac{n}{2} - z \right]$

$\sin(-a) = -\sin a$

$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$

$A \cap B$



Background

Problems:

Fragmented problem types

Inconsistent evaluation criteria

Difficulty comparing technologies

Background

Four pivotal dimensions:

- i) a comprehensive exploration of the various mathematical problems and their corresponding datasets that have been investigated
- ii) an examination of the spectrum of LLM-oriented techniques that have been proposed for mathematical problem-solving
- iii) an overview of factors and concerns affecting LLMs in solving math
- iv) an elucidation of the persisting challenges within this domain.

Math Problems & Datasets

-> Pure mathematical operations

-> Numerical manipulation

$$Q: 21 + 97$$

$$A: 118$$

Math Word Problems(MWP)

- > Mathematical exercises or scenarios
- > Written or verbal descriptions

Math Word Problems(MWP)

Question-Answer

Q: Lily received \$20 from her mum. After spending \$10 on a storybook and \$2.5 on a lollipop, how much money does she have left?

A: \$7.5

Question-Equation-Answer

Q: Jack had 8 pens and Mary had 5 pens. Jack gave 3 pens to Mary. How many pens does Jack have now?

E: $8 - 3$

A: 5 (optional)

Question-Rationale-Answer

Q: Beth bakes 4, or 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

R: Beth bakes 4 2 dozen batches of cookies for a total of $4 * 2 = << 4 * 2 = 8 >>$ 8 dozen cookies. There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 * 8 = << 12 * 8 = 96 >>$ 96 cookies. She splits the 96 cookies equally amongst 16 people so they each eat $96 / 16 = << 96 / 16 = 6 >>$ 6 cookies.

A: 6

Math Word Problems(MWP)

Tabular MWP

BEADS	\$/KILOGRAM
heart-shaped	3
rectangular	2
spherical	2
oval	2

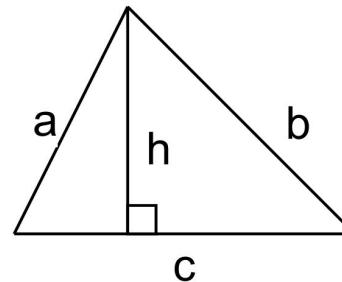
Table 2: Table for the tabular MWP example.

\mathcal{T} : Table 2

\mathcal{Q} : Henrik bought 2.5 kilograms of oval beads. How much did he spend? (Unit: \$)

\mathcal{A} : 5

- > Shapes
- > Sizes
- > Interrelationships



Q: $a=7$ inches; $b=24$ inches; $c=25$ inches;
 $h=6.72$ inches; What is its area? (Unit:
square inches)

A: 84

Math Problems & Datasets

	NAME	SIZE	LEVEL	NOTE
Q-A	CMATH (Wei et al., 2023) SAT-MATH (Zhong et al., 2023)	1.7K 220	E H	Chinese; grade 1-6 Multi-choice
Question-Equation-Answer	SVAMP (Patel et al., 2021)	1K	E	Three types of variations
	ASDIV (Miao et al., 2020)	2.3K	E	Problem type and grade level annotated
	MAWPS (Koncel-Kedziorski et al., 2016)	3.3K	E	Extension of ADDSUB, MULTIARITH, etc.
	PARAMAWPS (Raiyan et al., 2023)	16K	E	Paraphrased, adversarial MAWPS
	SINGLEEQ (Koncel-Kedziorski et al., 2015)	508	E	
	ADDSUB (Hosseini et al., 2014)	395	E	Only addition and subtraction
	MULTIARITH (Roy and Roth, 2015)	600	E	Multi-step reasoning
	DRAW-1K (Upadhyay and Chang, 2017)	1K	E	
	MATH23K (Wang et al., 2017)	23K	E	Chinese
	APE210K (Zhao et al., 2020)	210K	E	Chinese
	K6 (Yang et al., 2023)	600	E	Chinese; grade 1-6
	CM17K (Qin et al., 2021)	17K	M H	Chinese; grade 6-12
Question-Rationale-Answer	CARP (Zhang et al., 2023a)	4.9K	M	Chinese
	GSM8K (Cobbe et al., 2021)	8.5K	M	Linguistically diverse
	MATH (Hendrycks et al., 2021)	12.5K	H	Problems are put into difficulty levels 1-5
	PRM800K (Lightman et al., 2023)	12K	H	MATH w/ step-wise labels
	MATHQA (Amini et al., 2019)	37K	C	GRE examinations; have quality concern
	AQUA (Ling et al., 2017)	100K	C	GRE&GMAT questions
	ARB (Sawada et al., 2023)	105	C	Contest problems and university math proof
	GHOSTS (Frieder et al., 2023b)	709	C	
	THEOREMQA-MATH (Chen et al., 2023b)	442	C	Theorem as rationale
	LILA (Mishra et al., 2022)	132K	H	Incorporates 20 existing datasets
	MATH-INSTRUCT (Yue et al., 2023)	260K	H	Instruction-following style
	TABMWP (Lu et al., 2023b)	38K	H	Tabular MWP; below the College level

Table 1: Datasets for Math Word Problems.

E = Elementary, M = Middle School, H = High School, C = College, H = Hybrid

NAME	SIZE
GEOSHADE (Alvin et al., 2017)	102
GEOS (Seo et al., 2015)	186
GEOS++ (Sachan et al., 2017)	1.4K
GEOS-OS (Sachan and Xing, 2017)	2.2K
GEOMETRY3K (Lu et al., 2021)	3K
GEOQA (Chen et al., 2021a)	5K
UNIGEO (Chen et al., 2022)	14.5K

Table 3: Geometry datasets

Automated theorem proving & Math in vision-language context

-
- MINIF2F (Zheng et al., 2022): Evaluates systems (Metamath, Lean, Isabelle) on Olympiad-level problems.
 - HOList (Bansal et al., 2019): Tests sequential theorem proving using only preceding lemmas.
 - COQGYM (Yang & Deng, 2019): Provides 71K+ human-written proofs in Coq, enabling training and validation.
 - CHARTQA (Masry et al., 2022), with 9.6K human written questions and 23.1K model-generated questions have explored a variety of complex reasoning questions that involve several logical and arithmetic operations over charts.
 - MATHVISTA (Lu et al., 2023a): size: 6K; it features seven types of mathematical reasoning, and fine-grained meta data are available,

Grade School Math

Dataset Overview

- **Scale:** Contains about 8,500 math problems.
- **Language:** Both the problems and the answers are in English.
- **Applicable scenarios:** Training models to reason step by step and verifying mathematical logic capabilities.

Dataset structure

- **Question type:**
covers elementary school math knowledge points such as addition, subtraction, multiplication, division, fractions, percentages, geometry, and measurement.
- **Question format:**
Questions are described in natural language and are usually combined with daily scenarios (such as shopping, time calculation, allocation problems, etc.). The answer needs to be derived step by step, and finally a numerical result is obtained.

Grade School Math

Problem: Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

Solution: Beth bakes 4 2 dozen batches of cookies for a total of $4 \times 2 = <<4*2=8>>8$ dozen cookies

There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of $12 \times 8 = <<12*8=96>>96$ cookies

She splits the 96 cookies equally amongst 16 people so they each eat $96 / 16 = <<96/16=6>>6$ cookies

Final Answer: 6

Problem: Mrs. Lim milks her cows twice a day. Yesterday morning, she got 68 gallons of milk and in the evening, she got 82 gallons. This morning, she got 18 gallons fewer than she had yesterday morning. After selling some gallons of milk in the afternoon, Mrs. Lim has only 24 gallons left. How much was her revenue for the milk if each gallon costs \$3.50?

Mrs. Lim got 68 gallons - 18 gallons = $<<68-18=50>>50$ gallons this morning.

So she was able to get a total of 68 gallons + 82 gallons + 50 gallons = $<<68+82+50=200>>200$ gallons.

She was able to sell 200 gallons - 24 gallons = $<<200-24=176>>176$ gallons.

Thus, her total revenue for the milk is $\$3.50/\text{gallon} \times 176 \text{ gallons} = \$<<3.50*176=616>>616$.

Final Answer: 616

Problem: Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?

Solution: Tina buys 3 12-packs of soda, for $3 \times 12 = <<3*12=36>>36$ sodas

6 people attend the party, so half of them is $6 / 2 = <<6/2=3>>3$ people

Each of those people drinks 3 sodas, so they drink $3 \times 3 = <<3*3=9>>9$ sodas

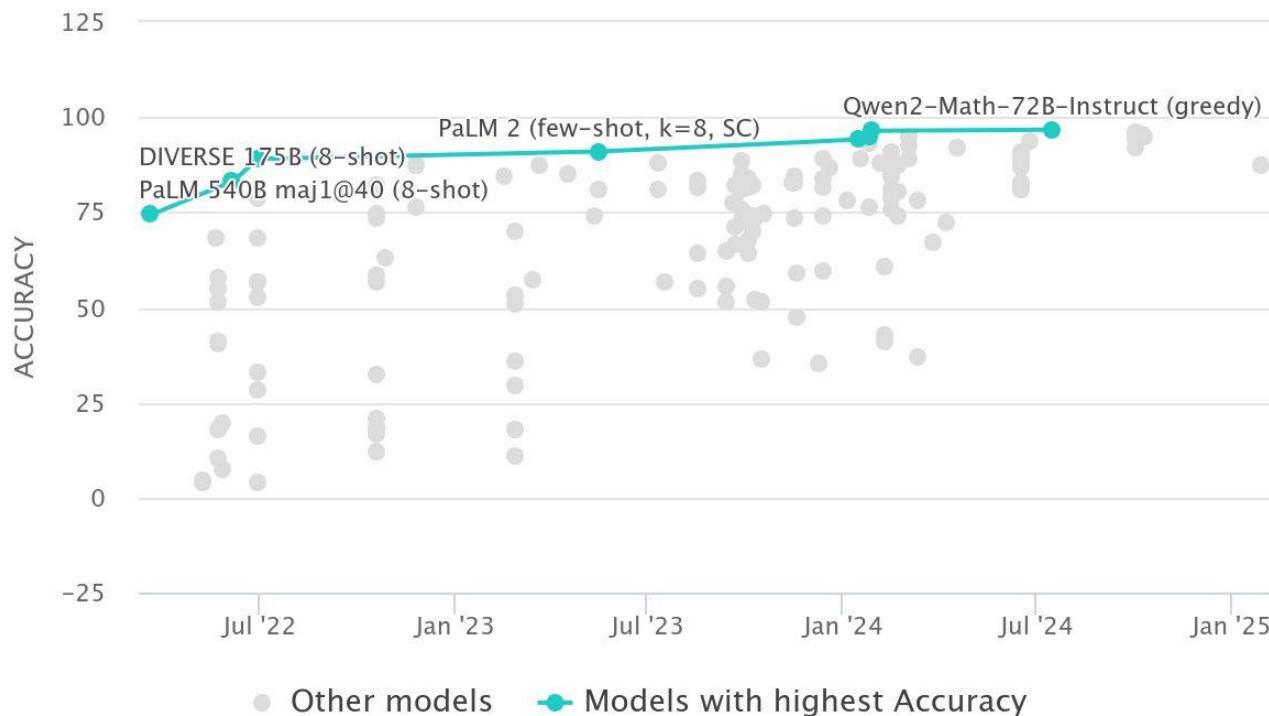
Two people drink 4 sodas, which means they drink $2 \times 4 = <<4*2=8>>8$ sodas

With one person drinking 5, that brings the total drank to $5 + 9 + 8 + 3 = <<5+9+8+3=25>>25$ sodas

As Tina started off with 36 sodas, that means there are $36 - 25 = <<36-25=11>>11$ sodas left

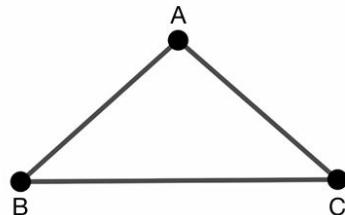
Final Answer: 11

Grade School Math



AlphaGeometry

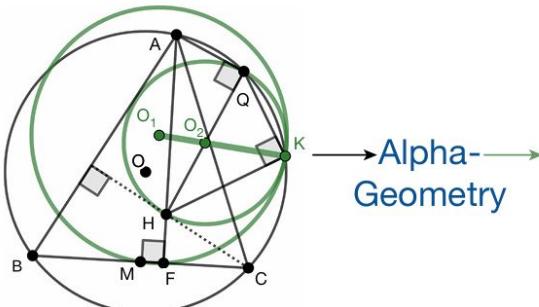
a A simple problem



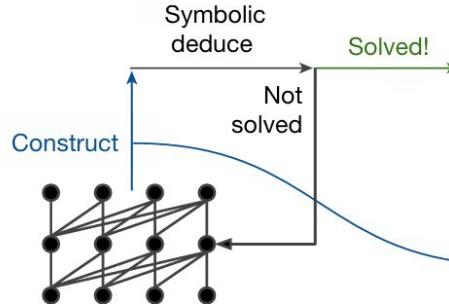
"Let ABC be any triangle with AB = AC.
Prove that $\angle ABC = \angle BCA$."

e IMO 2015 P3

"Let ABC be an acute triangle. Let (O) be its circumcircle, H its orthocenter, and F the foot of the altitude from A. Let M be the midpoint of BC. Let Q be the point on (O) such that $QH \perp QA$ and let K be the point on (O) such that $KH \perp KQ$. Prove that the circumcircles (O_1) and (O_2) of triangles FKM and KQH are tangent to each other."

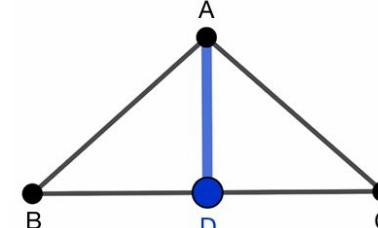


b AlphaGeometry



c Language model

d Solution



Construct D: midpoint BC,
 $AB=AC, BD = DC, AD=AD \Rightarrow \angle ABD=\angle DCA [1]$
 $[1], B C D \text{ collinear} \Rightarrow \angle ABC=\angle BCA$

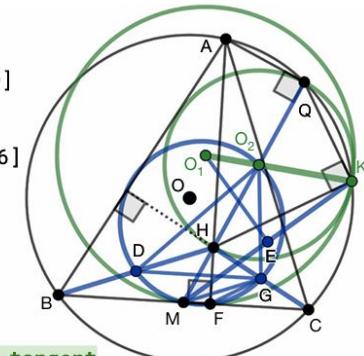
f Solution

Construct D: midpoint BH [a]
 $[a], O_2 \text{ midpoint HQ} \Rightarrow BQ \parallel O_2 D [20]$
 ...

Construct G: midpoint HC [b] ...
 $\angle GMD = \angle GO_2D \Rightarrow M O_2 G D \text{ cyclic} [26]$
 $[a], [b] \Rightarrow BC \parallel DG [30]$
 ...

Construct E: midpoint MK [c]
 ..., [c] $\Rightarrow \angle KFC = \angle KO_1E [104]$
 ...

$\angle FKQ = \angle FKQ_2 \Rightarrow KO_1 \parallel KO_2 [109]$
 $[109] \Rightarrow O_1 O_2 K \text{ collinear} \Rightarrow (O_1)(O_2) \text{ tangent}$



AlphaGeometry

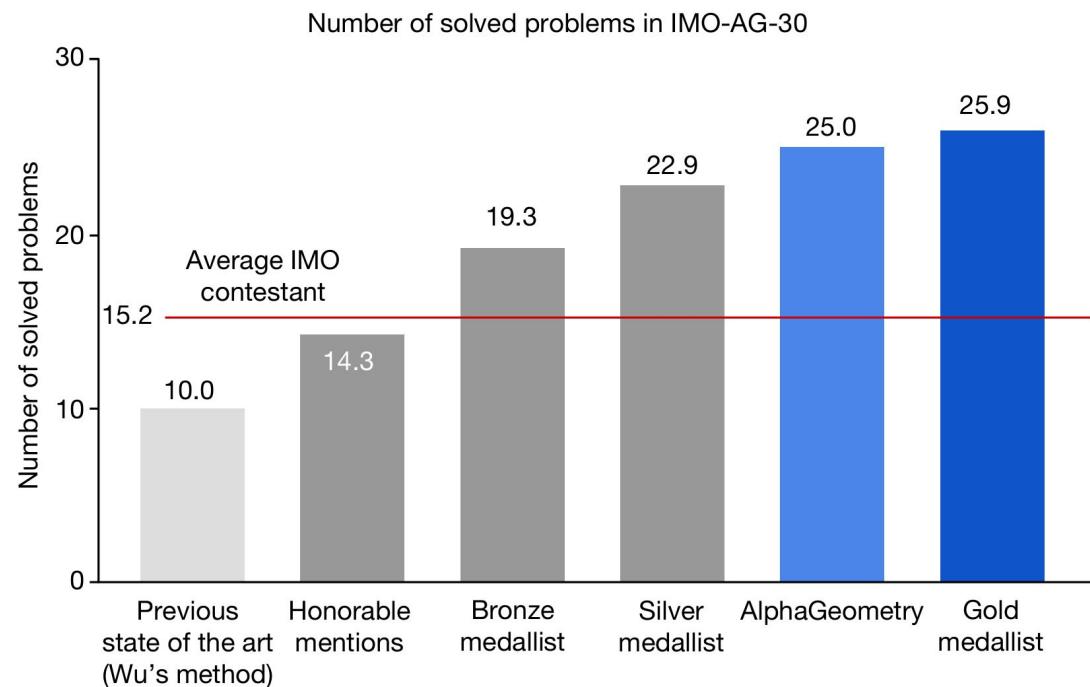


Table 1 | Main results on our IMO-AG-30 test benchmark

Method	Problems solved (out of 30)
Computer algebra	Wu's method ²¹ (previous state of the art)
	Gröbner basis ²⁰
Search (human-like)	GPT-4 (ref.25)
	Full-angle method ³⁰
Deductive database (DD) ¹⁰	7
DD+human-designed heuristics ¹⁷	9
DD+AR (ours)	14
DD+AR+GPT-4 auxiliary constructions	15
DD+AR+human-designed heuristics	18
AlphaGeometry	25
• Without pretraining	21
• Without fine-tuning	23

Related Work

Research Progress

Study	Research Focus	Math Domain Coverage	Educational Perspective	Human Factors Consideration
Frieder et al. (2023a)	ChatGPT version comparison Four theorem proving tasks	Theorem proving/Math search/Computation	None	Proposed human-AI collaboration
Chang et al. (2023)	General LLM evaluation	Math problem-solving (brief coverage)	None	None
Testolin (2023)	Deep learning & math reasoning	General math reasoning	None	None
Lu et al. (2023c)	Deep learning applications	Mathematical reasoning methodologies	None	None
Liu et al. (2023b)	LLM methods in mathematics	Multi-domain coverage	None	Not emphasized
This Paper	LLM-centric deep analysis	Comprehensive coverage	Yes	Emphasizes human factors

Matthew Nguyen (ttn5cv)



Methodologies

Overview of Methods

- Three progressive levels:
 - Prompting frozen LLMs
 - Strategies enhancing frozen LLMs
 - Fine-tuning LLMs
- Focus on improving math problem solving

Prompting Frozen LLMs

- Direct prompting with models like:
 - **GPT-3:** Used for classification, equation extraction, and question generation.
 - **ChatGPT:** Evaluated on MWP.
 - **GPT-4:** Explored with vanilla, Program-of-Thought, and Program Synthesis prompts.
 - **Multimodal Models:** GPT4V and Bard evaluated on visual contexts.

An Independent Evaluation of ChatGPT on MWP

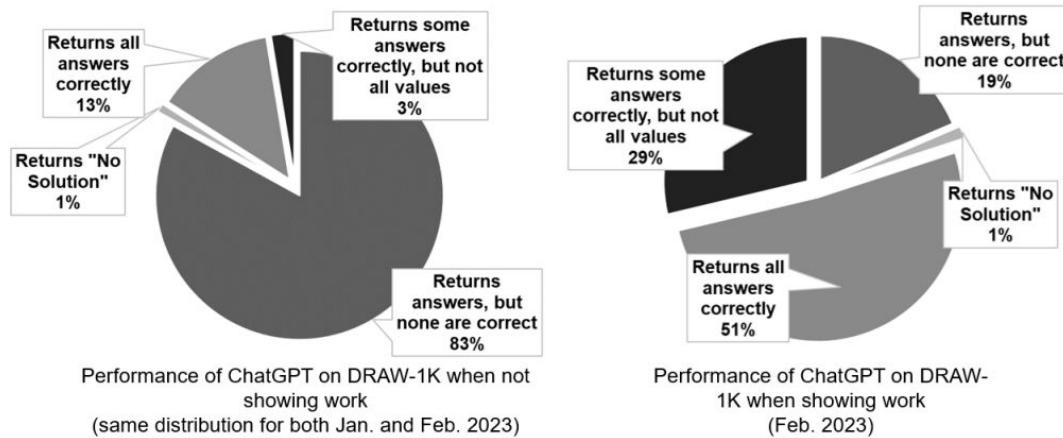


Figure 2: Overall results on the 1,000 MWPs in DRAW-1K based on ChatGPT's response.

An Independent Evaluation of ChatGPT on MWP

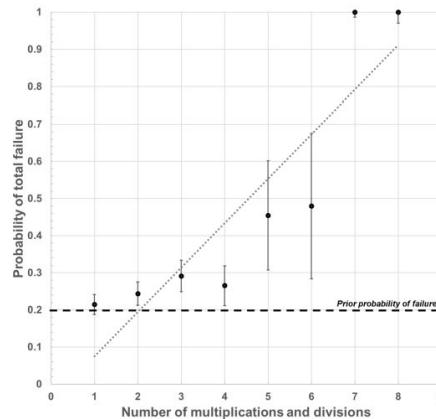


Figure 4: Additional finding specific to the February, 2023 experiment where ChatGPT displayed its work relating number of multiplications to probability of failure, $R^2 = 0.802$, 95% confidence intervals.

MATHVISTA: Evaluating Mathematical Reasoning Of Foundation Models In Visual Contexts

Model	Input	ALL	FQA	GPS	MWP	TQA	VQA	ALG	ARI	GEO	LOG	NUM	SCI	STA
<i>Heuristics baselines</i>														
<i>Large Language Models (LLMs)</i>														
Random chance	-	17.9	18.2	21.6	3.8	19.6	26.3	21.7	14.7	20.1	13.5	8.3	17.2	16.3
Frequent guess	-	26.3	22.7	34.1	20.4	31.0	24.6	33.1	18.7	31.4	24.3	19.4	32.0	20.9
<i>Augmented Large Language Models (Augmented-LLMs)</i>														
2-shot CoT Claude-2	Q only	23.5	21.9	26.9	9.1	38.6	23.5	27.7	15.9	25.7	21.6	9.9	41.5	20.5
Zero-shot GPT-4	Q only	26.1	22.3	37.0	7.0	39.2	27.4	33.6	17.4	35.6	16.2	9.2	45.8	19.5
Zero-shot Claude-2	Q only	26.4	21.9	34.1	13.4	36.1	29.1	32.8	20.4	33.3	13.5	12.1	36.4	20.5
2-shot CoT Claude-2	Q only	24.4	18.6	29.8	9.7	33.5	34.1	29.2	19.0	28.0	5.4	13.9	36.9	18.9
2-shot CoT ChatGPT	Q only	26.8	20.1	36.5	8.6	44.9	28.5	35.6	17.0	33.5	21.6	14.6	45.9	17.9
2-shot CoT GPT-4	Q only	29.2	20.1	44.7	8.6	46.2	31.3	41.6	19.3	41.0	18.9	13.9	47.5	18.9
2-shot PoT ChatGPT	Q only	25.1	19.0	30.8	16.1	38.0	25.7	29.9	19.8	29.3	24.3	19.4	38.5	16.9
2-shot PoT GPT-4	Q only	26.0	20.1	33.2	8.1	44.9	28.5	32.7	16.7	31.0	24.3	13.2	48.4	18.3
<i>Large Multimodal Models (LMMs)</i>														
IDEFICS-9B-Instruct	Q, I	19.8	21.6	21.1	6.5	25.9	24.0	22.1	15.0	19.8	18.9	9.9	24.6	18.1
mPLUG-Owl-LLaMA-7B	Q, I	22.2	22.7	23.6	10.2	27.2	27.9	23.6	19.2	23.9	13.5	12.7	26.3	21.4
miniGPT4-LLaMA-2-7B	Q, I	23.1	18.6	26.0	13.4	30.4	30.2	28.1	21.0	24.7	16.2	16.7	25.4	17.9
LLaMA-Adapter-V2-7B	Q, I	23.9	21.2	25.5	11.3	32.3	31.8	26.3	20.4	24.3	24.3	13.9	29.5	18.3
LLaVAR	Q, I	25.2	21.9	25.0	16.7	34.8	30.7	24.2	22.1	23.0	13.5	15.3	42.6	21.9
InstructBLIP-Vicuna-7B	Q, I	25.3	23.1	20.7	18.3	32.3	35.2	21.8	27.1	20.7	18.9	20.4	33.0	23.1
LLaVA-LLaMA-2-13B	Q, I	26.1	26.8	29.3	16.1	32.3	26.3	27.3	20.1	28.8	24.3	18.3	37.3	25.1
Multimodal Bard	Q, I	34.8	26.0	47.1	29.6	48.7	26.8	46.5	28.6	47.8	13.5	14.9	47.5	33.0
GPT-4V (Playground)	Q, I	49.9	43.1	50.5	57.5	65.2	38.0	53.0	49.0	51.0	21.6	20.1	63.1	55.8
<i>Human</i>														
Human performance	Q, I	60.3	59.7	48.4	73.0	63.2	55.9	50.9	59.2	51.4	40.7	53.8	64.9	63.9

Table 2: Accuracy scores on the *testmini* subset of MATHVISTA. Input: Q : question, I : image, I_c : image caption, I_t : OCR text detected in the image. ALL: overall accuracy. Task types: FQA: figure question answering, GPS: geometry problem solving, MWP: math word problem, TQA: textbook question answering, VQA: visual question answering. Mathematical reasoning types: ALG: algebraic reasoning, ARI: arithmetic reasoning, GEO: geometry reasoning, LOG: logical reasoning, NUM: numeric commonsense, SCI: scientific reasoning, STA: statistical reasoning. The highest scores among models in each section and overall are highlighted in blue and red, respectively.

Strategies Enhancing Frozen LLMs

- **Preprocessing:** Replacing numerical expressions with English words
- **Advanced Prompting:**
 - Self-Consistency techniques
 - Code-based self-verification (especially for GPT-4 Code Interpreter)
- **Using External Tools:** Python REPL and symbolic solvers
- **Interactive Frameworks:** e.g., MathChat for simulated conversation
- **Evaluation Beyond Accuracy:** Incorporating confidence levels and verifiable explanations

LPML: LLM-Prompting Markup Language For Mathematical Reasoning

- One of the most essential challenges in employing LLMs for mathematical reasoning is the management of calculation and reasoning errors in LLMs' outputs
- External tools like Python REPL have been used to handle calculations, but integration with CoT is not seamless.
- Reasoning and computation are often treated as separate modules, leading to inconsistent results.

LPML: LLM-Prompting Markup Language For Mathematical Reasoning

- Introduces LPML, an XML-like markup language that structures outputs (using tags like <THINK>, <PYTHON>, <OUTPUT>) for better control and parsing.
- Creates an interactive loop where the LLM generates both CoT reasoning and executable Python code, with a system feeding back verified results.
- Enhances accuracy by having the LLM prioritize computed results over its own internal reasoning.



LPML: LLM-Prompting Markup Language For Mathematical Reasoning

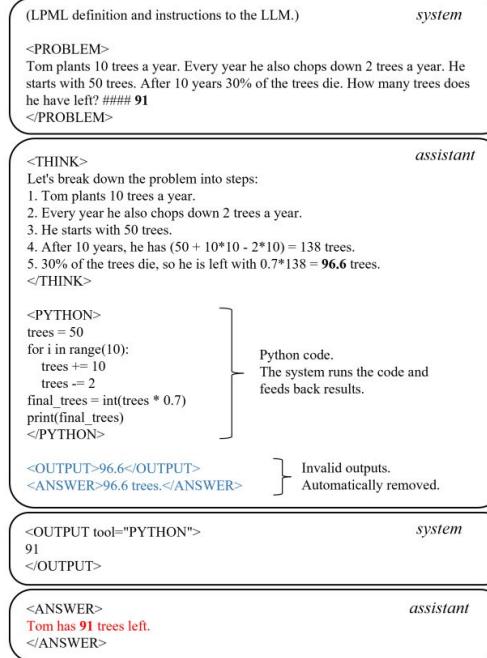
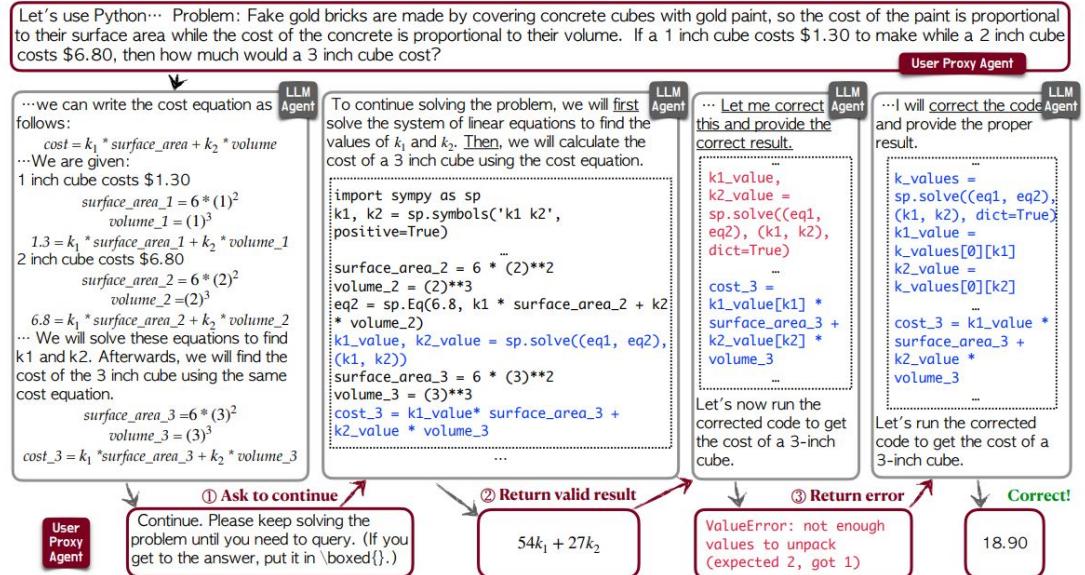


Figure 1: An example of the reasoning process: *assistant* (LLM) outputs CoT and Python code, while the *system* (computer) provides feedback on the code's execution results. Since *assistant* may output invalid elements, *system* removes them.

MathChat: Converse To Tackle Challenging Math Problems With LLM Agents

- While LPML is effective, the rigid markup can limit flexibility and natural dialogue flow in complex reasoning tasks.
- We need a more dynamic, interactive framework that adapts to iterative problem solving.
- MathChat adopts a conversational, multi-turn dialogue model where an LLM agent collaborates with a user proxy agent instead of a predefined system, leveraging the chat-optimized feature of state-of-the-art LLMs

MathChat: Converse To Tackle Challenging Math Problems With LLM Agents



Solving Challenging Math Word Problems Using GPT-4 Code Interpreter With Code-Based Self-Verification

- GPT-4 Code Interpreter is a variant of GPT-4 that integrates natural language reasoning with the capability to generate and execute code.
- The model is able to evaluate the outcomes of code execution and automatically adjust reasoning steps of solutions when needed.
- However, despite these advantages, GPT4-Code falls short in assuring final solution correctness.

Solving Challenging Math Word Problems Using GPT-4 Code Interpreter With Code-Based Self-Verification

- Explicit Code-Based Self-Verification (CSV): This method explicitly prompts the model to generate additional code dedicated to verifying its final answer.
- Iterative Correction: If the verification code indicates that the answer is incorrect (returns “False”), the model is prompted to revise and re-verify its solution.

$$C \rightarrow V = \begin{cases} \text{True} & \rightarrow \text{final answer} \\ \text{False} & \rightarrow C_{\text{new}} \rightarrow V \rightarrow \dots \rightarrow \text{True} \rightarrow \text{final answer} \\ \text{Uncertain} & \rightarrow \text{final answer} \end{cases}$$

Solving Challenging Math Word Problems Using GPT-4 Code Interpreter With Code-Based Self-Verification

Question

Question:
What is the smallest positive integer solution to $(\frac{100,000,000}{x^4}) < 1$?

Answer: 101

(a) Example: w/o self verification

Breaking this down into subproblems, the first step is to rearrange the inequality to isolate x . We can do this by multiplying both sides of the inequality by x^4 and then taking the fourth root of both sides. This gives us $x > \sqrt[4]{100000000}$. Let's solve for x .

code

```
import math
root = math.pow(100000000, 1/4)
>>> 100
```

Since 100 is already a positive integer, the smallest positive integer solution to the inequality $\frac{100,000,000}{x^4} < 1$ is $x = 100$.
The final answer is 100 X

(b) Example: w/ self verifications

We can firstly change the inequality into $x > \sqrt[4]{100000000}$. Then let's evaluate the fourth root of 100,000,000.

code

```
import math
root = math.pow(100000000, 1/4)
smallest_integer = math.ceil(root)
>>> (100.0, 100)
```

Thus, the smallest positive integer x is 100. Let's verify this solution.

code

```
verification = 1e8 / (smallest_integer ** 4) < 1
>>> False
```

The smallest integer x should be the next integer greater than 100, not 100 itself. Let's calculate that next integer.

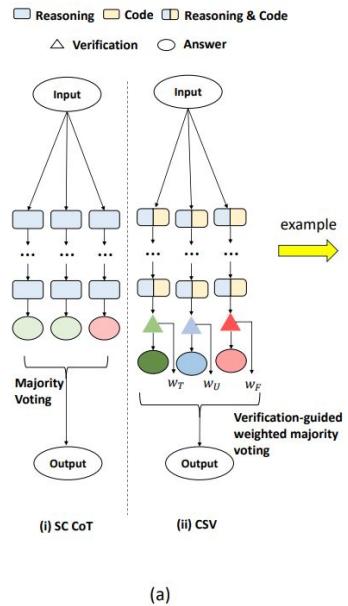
code

```
smallest_integer = 100 + 1
verification = 1e8 / (smallest_integer ** 4) < 1
>>> (101, True)
```

Upon substituting $x = 101$ back, it holds true, which verifies our solution.
The final answer is 101 ✓

Figure 3: Question from the 712th intermediate algebra problem in the MATH dataset. (a) Without self-verification, the model generates a wrong answer. (b) With self-verification, the model corrects the error and generates the correct answer. The CSV prompt: *To solve the problem using code interpreter step by step, and please verify your answer using code interpreter.*

Solving Challenging Math Word Problems Using GPT-4 Code Interpreter With Code-Based Self-Verification



Question: Given that the repetend in the decimal representation of $\frac{1}{19}$ contains 18 digits, find the 39th digit in the decimal representation.

Verified: True

Given that the sequence repeats every 18 digits, the 39th digit will be the same as the 3rd. So, we need to find the 3rd digit in the decimal representation of $\frac{1}{19}$

Code execution.....
The 3rd digit in the decimal representation of $\frac{1}{19}$ is 2. Now let's verify if this solution is equal to the 39th digit.

Code execution.....
given the repeating nature of the decimal representation of $\frac{1}{19}$, we can confidently state that the 39th digit will indeed be 2

Candidate answers:
2: 4
5: 6

Majority voting:

5 X

Verification-guided weighted majority voting:

3.5 > 2.3

2 ✓

Verified: Uncertain

If we calculate the first few decimal places of $\frac{1}{19}$, we will see a pattern emerge. Let's calculate this using Python.

Code execution.....
The 39th digit will be the same as the 21st, which is actually the 3rd digit in our calculated sequence. Let's find this digit.

Code execution.....
given the repeating nature of the decimal representation of $\frac{1}{19}$, we can confidently state that the 39th digit will indeed be 2

W_T = 1
W_U = 0.5
W_F = 0.2

True Uncertain False
(3, 1, 0)
score(2) = 1 × 3 + 0.5 × 1 + 0.2 × 0 = 3.5

5: 6
(1, 1, 4)
score(5) = 1 × 1 + 0.5 × 1 + 0.2 × 4 = 2.3

4 < 6

Verification-guided weighted majority voting:

3.5 > 2.3

5 X

2 ✓

Verified: False

Since the decimal representation repeats every 18 digits, the 39th digit will be the same as the 21st digit.

Code execution.....
The result is 0. Now let's verify the answer.
Code execution.....
The pattern of 18 repeating digits is "052631578947368421". The 21st digit, which is the same as the 39th digit, is '5', not '0'. The verification result is False. Our final result is 5

- Verification-Guided Voting: Combines multiple solution paths by assigning different weights based on their verification outcomes

Solving Challenging Math Word Problems Using GPT-4 Code Interpreter With Code-Based Self-Verification

Table 1: Accuracy (%) on MATH dataset. **VW-voting** is the abbreviation for the verification-guided weighted majority voting. (**Overall:** The results across various MATH subtopics (Hendrycks et al., 2021))

	Code-based Verification	VW-Voting	Intermediate Algebra	Precalculus	Geometry	Number Theory	Counting & Probability	PreAlgebra	Algebra	Overall MATH
GPT-4	✗	✗	-	-	-	-	-	-	-	42.20
GPT-3.5	✗	✗	14.6	16.8	22.3	33.4	29.7	53.8	49.1	34.12
GPT-4 (CoT)	✗	✗	23.4	26.7	36.5	49.6	53.1	71.6	70.8	50.36
GPT-4 (PHP)	✗	✗	26.3	29.8	41.9	55.7	56.3	73.8	74.3	53.90
GPT4-Code	✗	✗	50.1	51.5	53.4	77.2	70.6	86.3	83.6	69.69
GPT4-Code + CSV	✓	✗	56.6	53.9	54.0	85.6	77.3	86.5	86.9	73.54
<i>Improvement</i>			+6.5	+2.4	+0.6	+7.6	+6.7	+0.2	+3.3	+3.85
GPT4-Code + CSV + Voting	✓	✓ (k=16)	74.4	67.8	64.9	94.1	89.0	91.6	95.6	84.32
<i>Improvement</i>			+24.3	+16.3	+11.5	+16.9	+18.4	+5.3	+12.0	+14.63

Fine-tuning LLMs

- **Selecting In-Context Examples:** e.g., PROMPTPG learns which examples work best
- **Generating Intermediate Steps:** “Scratchpad” approaches for step-by-step reasoning
- **Answer Verifiers:** Fine-tuning models to assess their own solutions (pseudo-dual learning)
- **Enhanced Datasets & Knowledge Distillation:**
 - Training on error-correction pairs
 - Teacher-student frameworks
- **Solver Ensembles:** Combining multiple approaches for robust performance

Training Verifiers to Solve Math Word Problems

- **Generation:** First, a generator model (finetuned on the GSM8K dataset) is used to produce multiple candidate solutions for a given problem.
- **Verification:** A separate verifier model is then trained to assess the correctness of these candidate solutions. The verifier judges each solution (either at the full-solution level or at each token, with token-level predictions found to be more effective) based solely on whether the final answer is correct.

Training Verifiers to Solve Math Word Problems

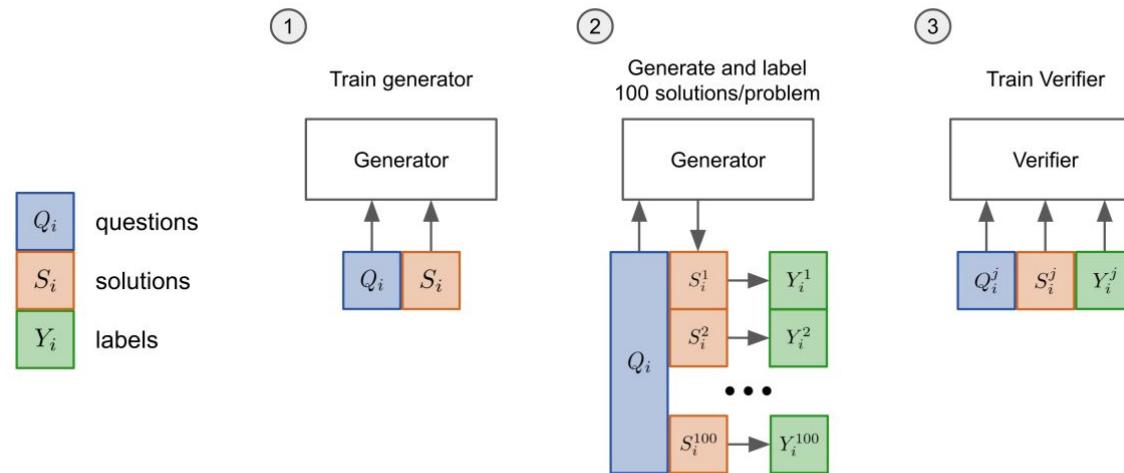


Figure 4: A diagram of the verification training pipeline.

Training Verifiers to Solve Math Word Problems

- The verifier is trained using a joint objective: it learns both to predict correctness (using a mean squared error loss on a scalar value for each token) and to perform language modeling.
- By sampling many solutions (typically 100 per problem) and labeling them as correct or incorrect, the verifier learns to rank candidate solutions reliably.

Training Verifiers to Solve Math Word Problems

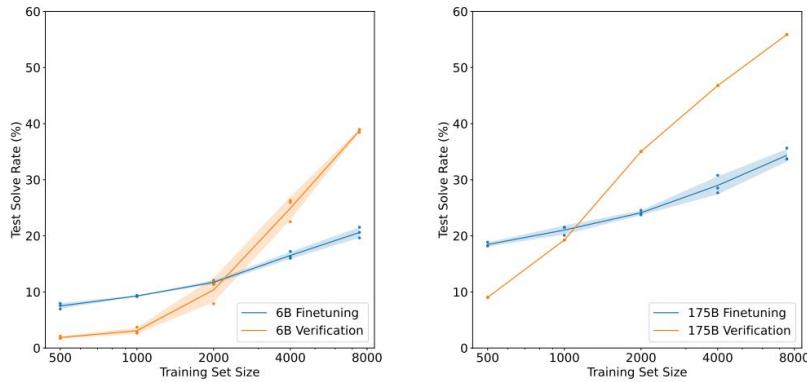


Figure 5: A comparison between finetuning and verification using 6B and 175B model sizes. Verification considers 100 solutions per problem. Mean and standard deviation is shown across 3 runs, except for 175B verification which shows only a single run.

Challenges, Analysis, and Implications

Challenges, Analysis, and Implications

- **Robustness & Vulnerabilities:** While instruction-tuned LLMs (e.g., GPT-4) have enhanced sensitivity and can maintain robustness even against distractions, they still struggle with complex or adversarially modified math problems, highlighting inherent vulnerabilities.
- **Critical Influencing Factors:** Key elements such as tokenization strategies, pre-training content (including code and LATEX), prompt design, and model scale fundamentally determine LLMs' arithmetic and reasoning performance.
- **Educational Implications:** Beyond raw problem-solving, LLMs impact math education by providing detailed, conversational, and step-by-step solutions that foster critical thinking, yet they also risk misinterpreting student needs and overcomplicating explanations, which can hinder effective learning.

Conclusion

Conclusion & Future Directions

- **Comprehensive Overview:** The survey reviews the landscape of large language models in mathematical reasoning, covering various types of math problems, associated datasets, and inherent challenges in the domain.
- **Advancements and Limitations:** It highlights recent progress in LLMs—including their improved problem-solving capabilities and applications in educational contexts—while also noting the current limitations and vulnerabilities of these models.
- **Future Directions:** The authors advocate for a more human-centric approach in math education and call for continued research to address persistent challenges and expand the practical applications of LLMs in diverse mathematical settings.

DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

Zeqiang Ning (avr7qy)



Introduction to DeepSeekMath



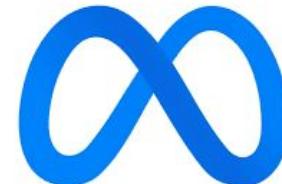
- Background
LLMs have revolutionized mathematical reasoning, but current open-source models fall short compared to cutting-edge models like GPT-4 and Gemini-Ultra, but DeepSeekMath outperform open-source models in math capabilities

Contributions

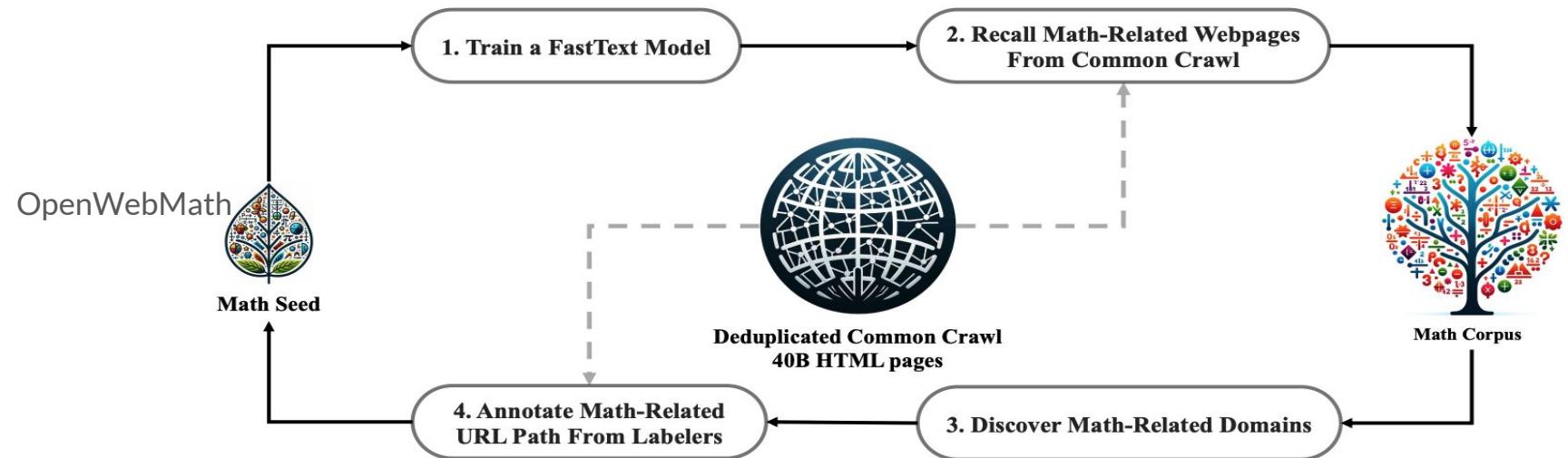
- Math Pre-Training at Scale
 - DeepSeekMath Corpus: 120B tokens, 7x Minerva, 9x OpenWebMath.
 - DeepSeekMath-Base 7B: Performs comparably to Minerva540B, showing data quality is key.
 - Code Training: Improves math problem-solving, with or without tools.
 - arXiv Training: No significant improvement in math benchmarks.

Data Collection—DeepSeekMath

- Construct a large-scale mathematical corpus from Common Crawl
- Approach: Iterative pipeline starting with a seed corpus
- FastText Model



Pipeline collecting data





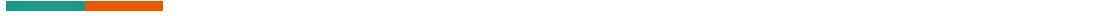
Validating the Data Quality

Math Corpus Comparison

DeepSeekMath	MathPile	OpenWebMath	Proof-Pile-2
120.2B	8.9B	13.6B	51.9B

Training Set

- Model: DeepSeekLLM 1.3B
- Training 150B tokens per corpus
- Optimizer: AdamW
- Batch size: 4M tokens
- Learning rate:
Warm-up for 2,000 steps
Decrease to 31.6% after 80% of training
Further decrease to 10.0% after 90% of training

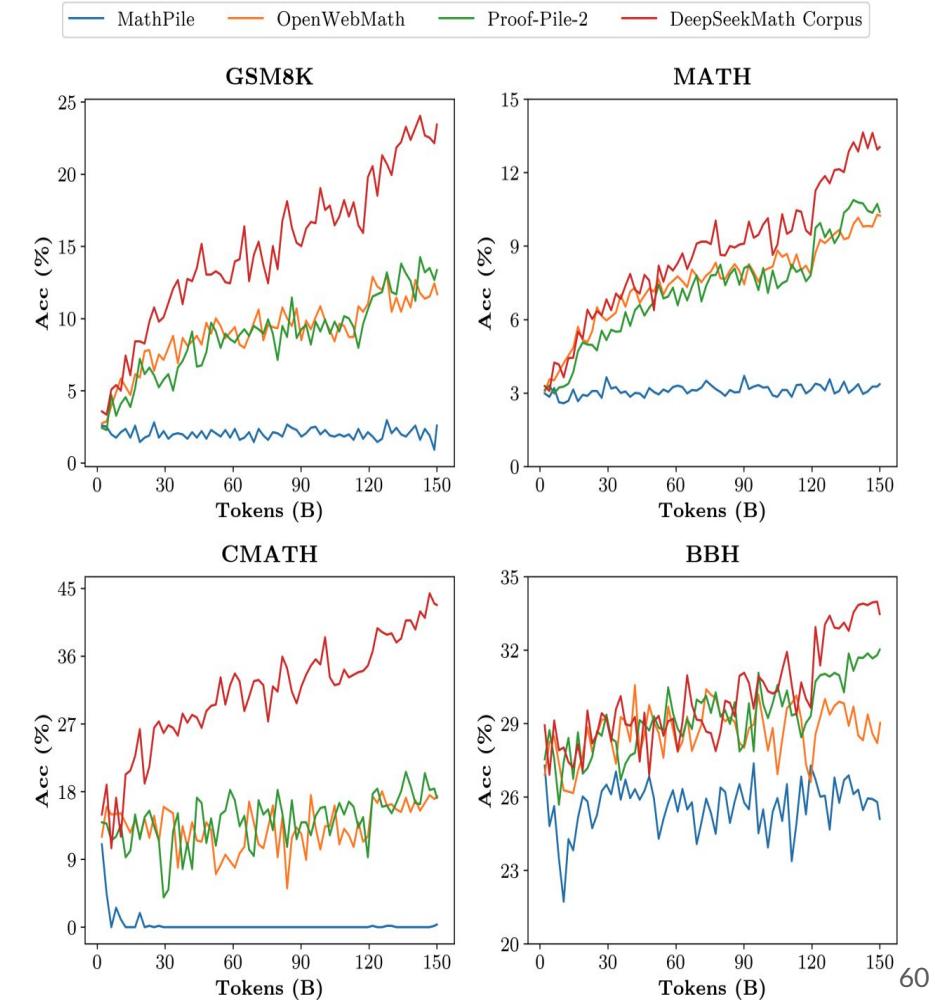


Evaluation of Corpus Results

Math Corpus	Size	English Benchmarks					Chinese Benchmarks		
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
No Math Training	N/A	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
MathPile	8.9B	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
OpenWebMath	13.6B	11.5%	8.9%	3.7%	31.3%	29.6%	16.8%	0.0%	14.2%
Proof-Pile-2	51.9B	14.3%	11.2%	3.7%	43.8%	29.2%	19.9%	5.1%	11.7%
DeepSeekMath Corpus	120.2B	23.8%	13.6%	4.8%	56.3%	33.1%	41.5%	5.9%	23.6%

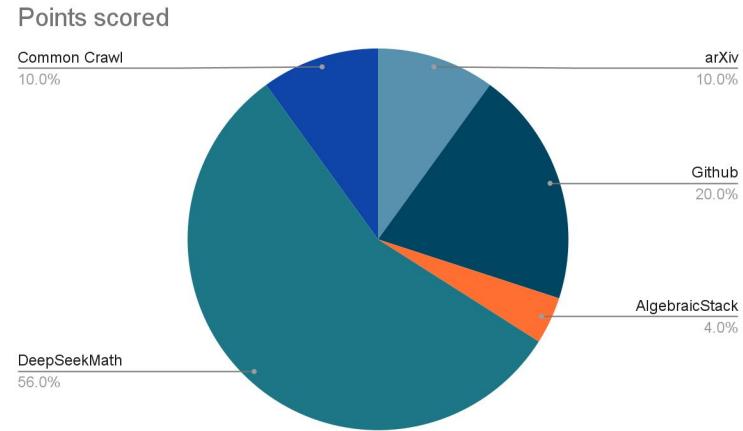
Evaluation of Corpus Results

- High-quality: few-shot chain-of-thought prompting
- Multilingual: Chinese and English
- Large-scale



Training DeepSeekMath-Base

- **Model:** Initialized with DeepSeek-Coder-Base-v1.5 7B, trained on 500B tokens.
- Data Distribution
- Capabilities:
 - Problem-solving with tools
 - Formal theorem proving
 - Natural language understanding
 - Reasoning and programming skills





Evaluating on Mathematical Problem Solving

Model	Size	English Benchmarks				Chinese Benchmarks			
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
Closed-Source Base Model									
Minerva	7B	16.2%	14.1%	7.7%	-	35.6%	-	-	-
Minerva	62B	52.4%	27.6%	12.0%	-	53.9%	-	-	-
Minerva	540B	58.8%	33.6%	17.6%	-	63.9%	-	-	-
Open-Source Base Model									
Mistral	7B	40.3%	14.3%	9.2%	71.9%	51.1%	44.9%	5.1%	23.4%
Llemma	7B	37.4%	18.1%	6.3%	59.4%	43.1%	43.4%	11.9%	23.6%
Llemma	34B	54.0%	25.3%	10.3%	71.9%	52.9%	56.1%	11.9%	26.2%
DeepSeekMath-Base	7B	64.2%	36.2%	15.4%	84.4%	56.5%	71.7%	20.3%	35.3%

Evaluating on Mathematical Problem Solving

Model	Size	Problem Solving w/ Tools		Informal-to-Formal Proving	
		GSM8K+Python	MATH+Python	miniF2F-valid	miniF2F-test
Mistral	7B	48.5%	18.2%	18.9%	18.0%
CodeLlama	7B	27.1%	17.2%	16.3%	17.6%
CodeLlama	34B	52.7%	23.5%	18.5%	18.0%
Llemma	7B	41.0%	18.6%	20.6%	22.1%
Llemma	34B	64.6%	26.3%	21.0%	21.3%
DeepSeekMath-Base	7B	66.9%	31.4%	25.8%	24.6%

Evaluating on Natural Language

Model	Size	MMLU	BBH	HumanEval (Pass@1)	MBPP (Pass@1)
Mistral	7B	62.4%	55.7%	28.0%	41.4%
DeepSeek-Coder-Base-v1.5 [†]	7B	42.9%	42.9%	40.2%	52.6%
DeepSeek-Coder-Base-v1.5	7B	49.1%	55.2%	43.2%	60.4%
DeepSeekMath-Base	7B	54.9%	59.5%	40.9%	52.6%

DeepSeekMath-Base 7B significantly outperforms DeepSeek-Coder-Base-v1.5 on MMLU, BBH, and coding benchmarks (HumanEval and MBPP), and surpasses the general model Mistral 7B, demonstrating the positive impact of math training on language understanding, reasoning, and coding abilities.

Supervised Fine-Tuning

- Constructing a mathematical instruction-tuning dataset covering English and Chinese problems from different mathematical fields and of varying complexity levels.
- **DeepSeekMath-Instruct 7B** is a model that undergoes mathematical instruction tuning based on DeepSeekMath-Base and a mathematical instruction tuning dataset
 - evaluating on four quantitative reasoning benchmarks
 - Comparing with leading models.

Evaluating

1. In the evaluation where tool use is disallowed, DeepSeekMath-Instruct 7B surpasses all open-source models and most proprietary models (e.g., Inflection-2 and Gemini Pro) on the MATH dataset, but still underperforms GPT-4 and Gemini Ultra.
2. In the evaluation where tool use is allowed, DeepSeekMath-Instruct 7B achieves an accuracy of nearly 60% on MATH, surpassing all open-source models and competing with DeepSeek-LLM-Chat .

Model	Size	English Benchmarks		Chinese Benchmarks		
		GSM8K	MATH	MGSM-zh	CMATH	
Chain-of-Thought Reasoning						
Closed-Source Model						
Gemini Ultra	-	94.4%	53.2%	-	-	
GPT-4	-	92.0%	52.9%	-	86.0%	
Inflection-2	-	81.4%	34.8%	-	-	
GPT-3.5	-	80.8%	34.1%	-	73.8%	
Gemini Pro	-	86.5%	32.6%	-	-	
Grok-1	-	62.9%	23.9%	-	-	
Baichuan-3	-	88.2%	49.2%	-	-	
GLM-4	-	87.6%	47.9%	-	-	
Open-Source Model						
InternLM2-Math	20B	82.6%	37.7%	-	-	
Qwen	72B	78.9%	35.2%	-	-	
Math-Shepherd-Mistral	7B	84.1%	33.0%	-	-	
WizardMath-v1.1	7B	83.2%	33.0%	-	-	
DeepSeek-LLM-Chat	67B	84.1%	32.6%	74.0%	80.3%	
MetaMath	70B	82.3%	26.6%	66.4%	70.9%	
SeaLLM-v2	7B	78.2%	27.5%	64.8%	-	
ChatGLM3	6B	72.3%	25.7%	-	-	
WizardMath-v1.0	70B	81.6%	22.7%	64.8%	65.4%	
DeepSeekMath-Instruct	7B	82.9%	46.8%	73.2%	84.6%	
DeepSeekMath-RL	7B	88.2%	51.7%	79.6%	88.8%	
Tool-Integrated Reasoning						
Closed-Source Model						
GPT-4 Code Interpreter	-	97.0%	69.7%	-	-	
Open-Source Model						
InternLM2-Math	20B	80.7%	54.3%	-	-	
DeepSeek-LLM-Chat	67B	86.7%	51.1%	76.4%	85.4%	
ToRA	34B	80.7%	50.8%	41.2%	53.4%	
MAmmoTH	70B	76.9%	41.8%	-	-	
DeepSeekMath-Instruct	7B	83.7%	57.4%	72.0%	84.3%	
DeepSeekMath-RL	7B	86.7%	58.8%	78.4%	87.6%	

Wenhao Xu (wx8mcm)



Reinforcement Learning

Reinforcement Learning Intro

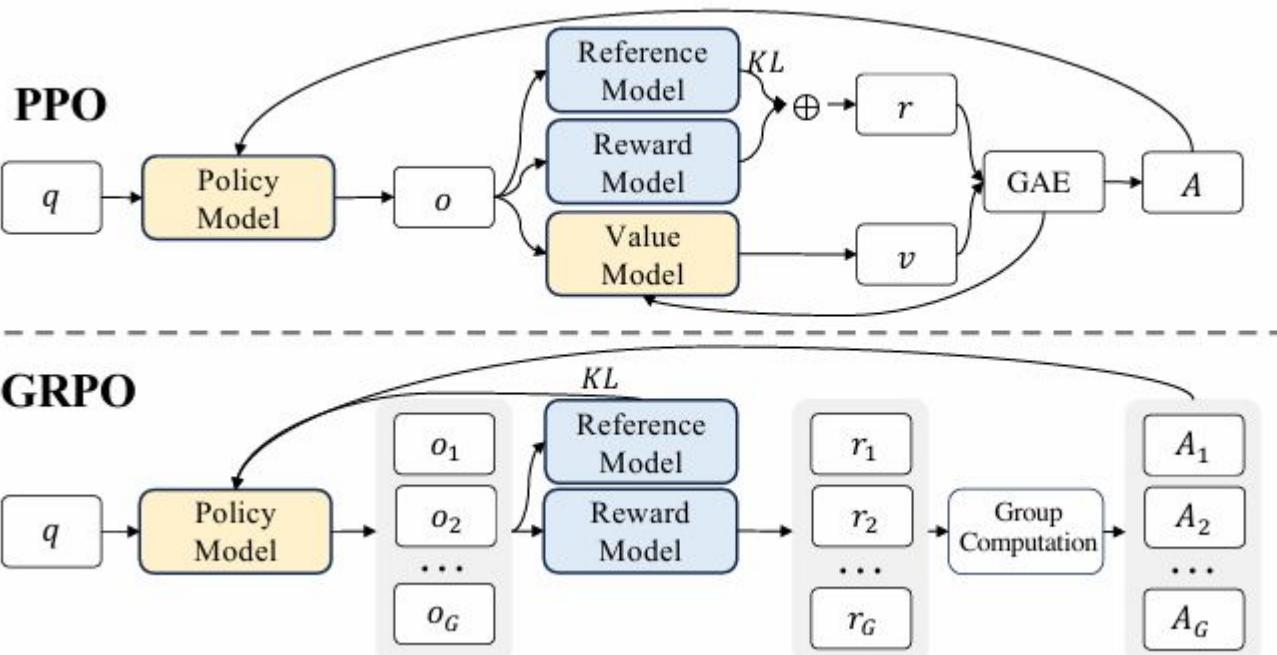
- Purpose of RL Post-SFT
 - Enhance model reasoning abilities beyond supervised training limits.
- Reinforcement Learning Phases
 - Fine-tuning through iterative feedback and reward-based optimization.
- In-Domain vs. Out-of-Domain Tasks
 - RL improves performance on both familiar and new benchmarks.

From PPO to GRPO

- PPO uses actor-critic models, high resource usage.
- GRPO eliminates the critic model.
- Baseline estimated from group scores.
- Reduces training resources significantly.

GRPO Methodology

- Samples multiple outputs per question.
- Uses average reward as baseline.
- Regularizes with KL divergence between policy and reference models.



GRPO vs PPO

- Computational Efficiency
 - GRPO significantly reduces memory requirements compared to PPO.
- Performance Boosts
 - GRPO led to improvements from 46.8% to 51.7% on MATH benchmark.
- Unified Paradigm for RL Techniques
 - GRPO fits into a broader framework of reinforcement learning strategies like RFT and DPO.

Training Process

- Outcome Supervision RL
- Process Supervision RL
- Iterative RL

Evaluation

- Benchmarked against leading models (GPT-4, Gemini Ultra, etc.).
- Without Tool Use:
 - Surpasses all open-source models on MATH.
 - Outperforms many proprietary models.
- With Tool Use:
 - Approaches 60% accuracy on MATH.
 - Competitive with larger models like DeepSeek-LLM-Chat 67B.

Model	Size	English Benchmarks		Chinese Benchmarks	
		GSM8K	MATH	MGSMS-zh	CMATH
Closed-Source Model					
Gemini Ultra	-	94.4%	53.2%	-	-
GPT-4	-	92.0%	52.9%	-	86.0%
Inflection-2	-	81.4%	34.8%	-	-
GPT-3.5	-	80.8%	34.1%	-	73.8%
Gemini Pro	-	86.5%	32.6%	-	-
Grok-1	-	62.9%	23.9%	-	-
Baichuan-3	-	88.2%	49.2%	-	-
GLM-4	-	87.6%	47.9%	-	-
Open-Source Model					
InternLM2-Math	20B	82.6%	37.7%	-	-
Qwen	72B	78.9%	35.2%	-	-
Math-Shepherd-Mistral	7B	84.1%	33.0%	-	-
WizardMath-v1.1	7B	83.2%	33.0%	-	-
DeepSeek-LLM-Chat	67B	84.1%	32.6%	74.0%	80.3%
MetaMath	70B	82.3%	26.6%	66.4%	70.9%
SeaLLM-v2	7B	78.2%	27.5%	64.8%	-
ChatGLM3	6B	72.3%	25.7%	-	-
WizardMath-v1.0	70B	81.6%	22.7%	64.8%	65.4%
DeepSeekMath-Instruct	7B	82.9%	46.8%	73.2%	84.6%
DeepSeekMath-RL	7B	88.2%	51.7%	79.6%	88.8%
Tool-Integrated Reasoning					
Closed-Source Model					
GPT-4 Code Interpreter	-	97.0%	69.7%	-	-
Open-Source Model					
InternLM2-Math	20B	80.7%	54.3%	-	-
DeepSeek-LLM-Chat	67B	86.7%	51.1%	76.4%	85.4%
ToRA	34B	80.7%	50.8%	41.2%	53.4%
MAmmoTH	70B	76.9%	41.8%	-	-
DeepSeekMath-Instruct	7B	83.7%	57.4%	72.0%	84.3%
DeepSeekMath-RL	7B	86.7%	58.8%	78.4%	87.6%

Discussion

Pre-Training Insights

- Code Training Benefits:
- Enhances mathematical reasoning both with and without tool use.
- Mixed code/math training mitigates catastrophic forgetting.
- Two-stage training: Code followed by math training yields best results.

Impact of Code Training

- Code training boosts program-aided mathematical reasoning.
- Enhances efficiency of subsequent math training.
- Mixed training improves reasoning and coding performance.

Training Setting	Training Tokens			w/o Tool Use			w/ Tool Use	
	General	Code	Math	GSM8K	MATH	CMATH	GSM8K+Python	MATH+Python
No Continual Training	-	-	-	2.9%	3.0%	12.3%	2.7%	2.3%
Two-Stage Training								
Stage 1: General Training	400B	-	-	2.9%	3.2%	14.8%	3.3%	2.3%
Stage 2: Math Training	-	-	150B	19.1%	14.4%	37.2%	14.3%	6.7%
Stage 1: Code Training	-	400B	-	5.9%	3.6%	19.9%	12.4%	10.0%
Stage 2: Math Training	-	-	150B	21.9%	15.3%	39.7%	17.4%	9.4%
One-Stage Training								
Math Training	-	-	150B	20.5%	13.1%	37.6%	11.4%	6.5%
Code & Math Mixed Training	-	400B	150B	17.6%	12.1%	36.3%	19.7%	13.5%

ArXiv Papers and Mathematical Reasoning

- Limited improvement from arXiv paper pre-training.
- No notable gains on GSM8K, MATH, and other benchmarks.
- Potential factors:
 - ArXiv content may not align with problem-solving tasks.
 - Impact may vary with model scale or specific tasks.

Model	Size	ArXiv Corpus	English Benchmarks					Chinese Benchmarks		
			GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
DeepSeek-LLM	1.3B	No Math Training	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
		MathPile	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
		ArXiv-RedPajama	3.3%	3.4%	4.0%	9.4%	9.0%	7.4%	0.8%	2.3%
DeepSeek-Coder-Base-v1.5	7B	No Math Training	29.0%	12.5%	6.6%	40.6%	38.1%	45.9%	5.9%	21.1%
		MathPile	23.6%	11.5%	7.0%	46.9%	35.8%	37.9%	4.2%	25.6%
		ArXiv-RedPajama	28.1%	11.1%	7.7%	50.0%	35.2%	42.6%	7.6%	24.8%

Conclusion, Future Work

Conclusion

- DeepSeekMath significantly outperforms all open-source models on competition-level MATH benchmarks.
- Approaches the performance of leading closed-source models like GPT-4 and Gemini-Ultra.
- Key Findings:
 - Public web data can serve as a high-quality resource for mathematical reasoning.
 - Code training prior to math training enhances reasoning capabilities.
 - Group Relative Policy Optimization (GRPO) improves reasoning with optimized memory usage.

Limitations

- DeepSeekMath underperforms in geometry and formal theorem proving compared to closed-source models.
- Struggles with problems involving specific geometric shapes like triangles and ellipses.
- Model scale limitations hinder few-shot learning capabilities.
- Reliance on publicly available data may introduce quality and coverage gaps

Future Work

- Enhancing RL Techniques
 - Further refining GRPO and exploring hybrid RL approaches for better performance.
- Expanding Multilingual Datasets
 - Incorporate more languages to broaden model applicability in global benchmarks.
- Combining Code and Math Training
 - Explore deeper integration of code and math data to enhance both reasoning and computational skills.

Questions?

Thank you!
