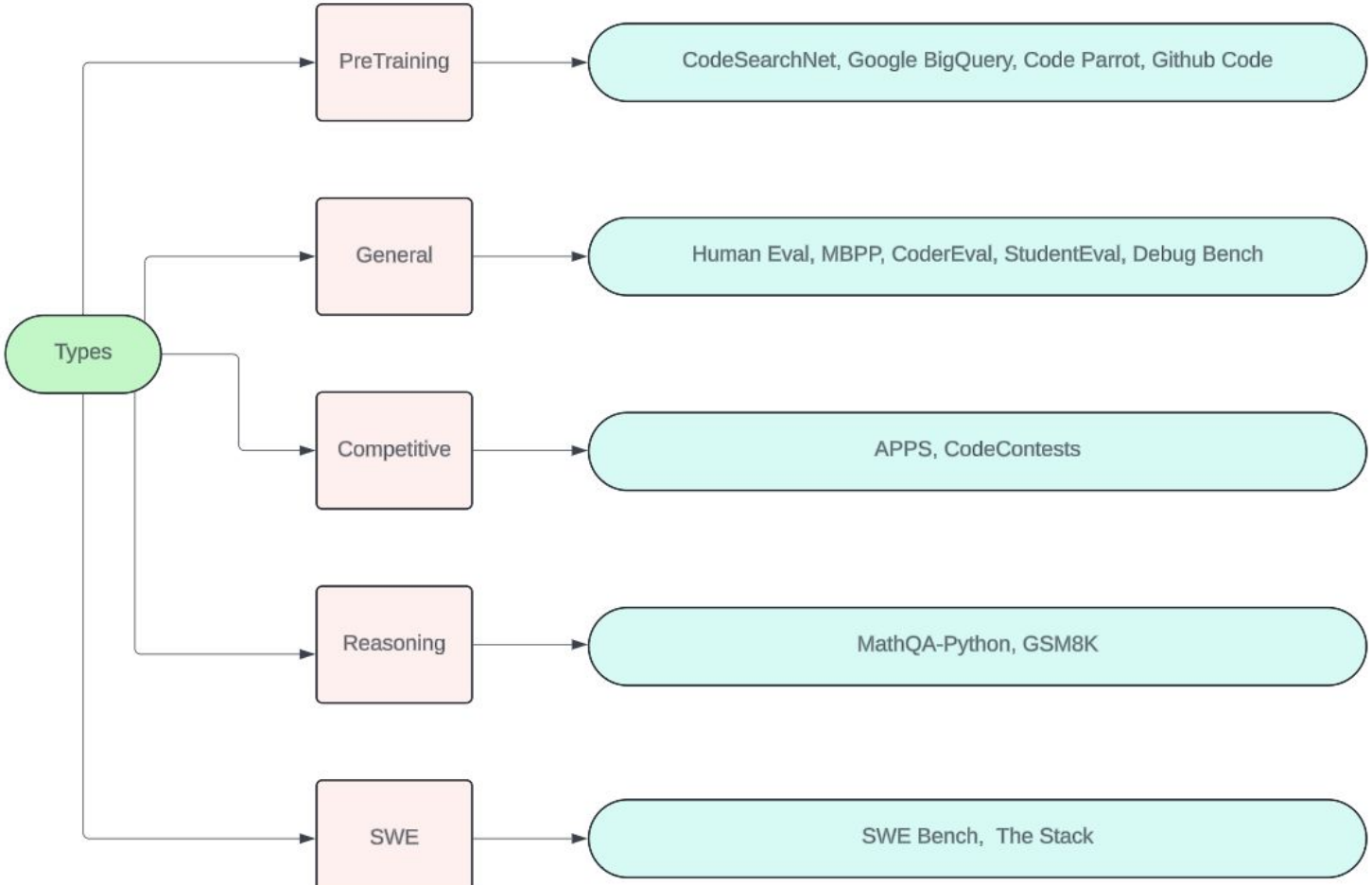
# Towards Effective Reasoning

By Chaitanya Garg(2021248)

# Motivation

Try to make a dynamic solving paradigm using dynamic testing
Statistic and Dynamic Test Generation having a programmer and a Tester

# Data Set Overview



# MATHPILE : A Billion-Token-Scale Pre-training Corpus for Math

By Yuxuan Tong

# Introduction

-> High Quality and Clean datasets are key ingredients for building futuristic foundation models and improving emergent behavior.

-> Mathpile(creating a high-quality and diverse pre-training corpus tailored for the math domain), a math based dataset created by thorough cleaning following in the footsteps of the principle ‘less is more’. Steps included but not limited to preprocessing, prefiltering, language identification, cleaning, filtering, and deduplication, + contamination detection.

->This has to be while capturing the aspects of Math Centric, Diversity, High-Quality and Data Documentation.

# Design Overview

-> Math-English-Open Sourced-based: Earlier ones are generalised (DOLMA, REDPAJAMA, PILE ) or multilingual(ROOTS and THE STACK) or closed source(MathPix, Minerva)

-> Diversity: Rather than focusing on just a specific domain of problems all problems from K-12 levels, university levels and even research levels have been included to cover a wide variety of topics and their development from basics to advanced. Referring to a variety of sources like books, papers, webpages unlike OpenWebMath.

-> High Quality: Employing Rule Based Filtering to remove low quality data in the form of poor explanations, repeated questions which might spoil the purpose.

-> Properly Documented the sources for reference.

# Data Sources

Math TextBook: 38 K-12 textbook, 369 college books, 347 college lectures, and refined synthetic textbooks from the OpenPhi Project.

Research Papers based on 50 sub-subjects spanning Mathematics, Computer Science, Statistics, Physics, Quantitative Finance and Economics

Math Wiki: a collection of 106,881 mathematical Wikipedia entries, about 0.8 billion tokens

Proof Wiki & Math Exchange: Used web parsers to read and removed repeated entries then, refined those.

Math Web Pages from Common Crawl: Refined Pages to use from Common Crawl

# Data Sources of MATHPILE VS OTHERS

PS-> Problem Set

| Datasets | Open Source | Type | Target Domain | # Textbooks | Has Synth. Data | Data Contam. Detection | # Tokens | Source |
|---|---|---|---|---|---|---|---|---|
| Minerva | ✗ | Corpus | General Math | ✗ | ✗ | ✓ | 38.5B | arXiv, Web |
| MathMix | ✗ | Corpus + PS | General Math | ? | ✓ | ✓ | 1.5B | ? |
| ProofPile | ✓ | Corpus | Theorem Proving | 7 | ✗ | ✗ | 8.3B | arXiv, Textbooks, Lib., StackExchange, ProofWiki, MATH |
| OpenWebMath | ✓ | Corpus | General Math | ✗ | ✗ | ✗ | 14.7B | Web |
| DM-Mathematics | ✓ | PS | Math Competition | ✗ | ✓ | - | 4.4B | Synthesis |
| AMPS | ✓ | PS | Math Competition | ✗ | ✓ | ✗ | 0.7B | Khan Academy, Synthesis |
| MATHPILE (Ours) | ✓ | Corpus | General Math | 3,979 | ✓ | ✓ | 9.5B | arXiv, Textbooks, StackExchange, Wikipedia, ProofWiki, Web |

# Data Processing(1)

-> Language Identification: To check if the language of the content is english, the filtering models had trouble due to the excessive math content so threshold criterion was accordingly adjusted.

-> Data Clean+Filter:

detect lines with ‘loren ipsum’/ ‘javascript’

Filter lines with less than words and having key words, way too much uppercase/citation/non-alphabets char

Filter on basis of word size, stop words and ellipses.

# Data Processing(2)

-> Data Deduplication: Even if unique data is being collected from each source, there is a possibility that there are commonalities in the data. This similarity was checked for examples above a threshold were removed (about 714 million tokens)

-> Data Contamination: Aggregated Questions and searched for similar ones in MMLU, AGIEval, MathQA and AQuA.

# Math Pile Statistics(Distribution)

Table 3: The components and data statistics of MATHPILE .

| Components | Size (MB) | # Documents | # Tokens | max(# Tokens) | min (# Tokens) | ave (# Tokens) |
|---|---|---|---|---|---|---|
| Textbooks | 644 | 3,979 | 187,194,060 | 1,634,015 | 256 | 47,046 |
| Wikipedia | 274 | 22,795 | 59,990,005 | 109,282 | 56 | 2,632 |
| ProofWiki | 23 | 23,839 | 7,608,526 | 6,762 | 25 | 319 |
| CommonCrawl | 2,560 | 75,142 | 615,371,126 | 367,558 | 57 | 8,189 |
| StackExchange | 1,331 | 433,751 | 253,021,062 | 125,475 | 28 | 583 |
| arXiv | 24,576 | 343,830 | 8,324,324,917 | 4,156,454 | 20 | 24,211 |
| Total | 29,408 | 903,336 | 9,447,509,696 | - | - | 10,458 |

# Experiment Results

| Models | GSM8K | MATH | SAT-MATH | MMLU-Math | MathQA | AQuA |
|---|---|---|---|---|---|---|
| Mistral-7B-v0.1 | 47.38 | 10.08 | 47.27 | 44.92 | 23.51 | 27.95 |
| + Textbooks (0.56B) | **48.97** | **12.10** | **56.36** | **48.93** | **30.38** | **33.07** |
| + Wikipedia (0.18B) | **49.96** | 9.96 | **53.63** | **47.16** | **28.97** | **35.43** |
| + StackExchange (0.87B) | 43.06 | **11.66** | 47.27 | 43.51 | **27.67** | **30.70** |
| + Common Crawl (1.83B) | 45.56 | 9.88 | **50.45** | **45.17** | **25.79** | **31.88** |
| + arXiv (0.38B) | **47.91** | 7.50 | 42.72 | **46.34** | 18.05 | 27.55 |
| + Textbooks, Wikipeida, StackEx., CC (4B) | **49.88** | **11.70** | 43.18 | 43.75 | 23.24 | 25.19 |
| + AMPS (1B) | 0.08 | 0.82 | 3.18 | 0.47 | 10.99 | 8.27 |
| + DM-Mathematics (5B) | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| + Sampled OpenWebMath (0.59B) | 43.21 | 7.86 | **47.72** | **47.52** | 21.80 | 24.80 |

# Questions to Ponder

-> Wouldn’t it be better to integrate similar subjects like Physics, Economics and CS from high school and university books too than just adding research papers?

# DART-Math: Difficulty-Aware Rejection Tuning for Mathematical Problem Solving

By Yuxuan Tong

# Introduction

-> Difficulty Aware Rejection Tuning (DART) : To mitigate basis from easy queries

->Utilizing DART, they have created new datasets for mathematical problem-solving that focus more on difficult queries and are substantially smaller than previous ones

-> Focused on using Open Source Models rather than Proprietary ones like Chat-GPT

-> Synthesised 6 mathematical benchmark called DART-MATH

# Introduction

-> Difficulty Aware Rejection Tuning (DART) : To mitigate basis from easy queries

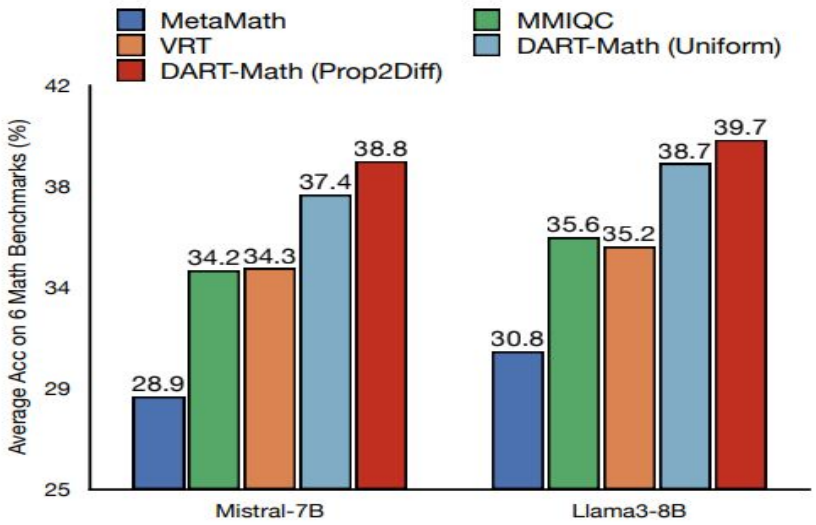
->Utilizing DART, they have created new datasets for mathematical problem-solving that focus more on difficult queries and are substantially smaller than previous ones

-> Focused on using Open Source Models rather than Proprietary ones like Chat-GPT

-> Synthesised 6 mathematical benchmark called DART-MATH

-> Develop cost-effective models

# Current Datasets



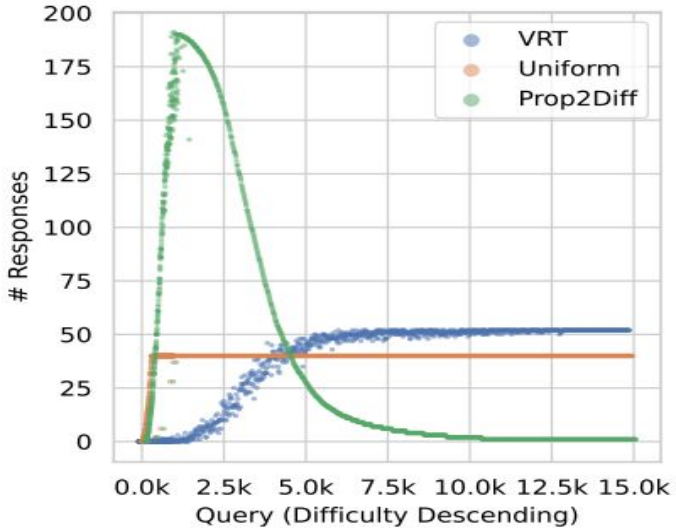




Figure 1: **Left:** Average accuracy on six mathematical benchmarks. We compare with models fine-tuned on the best, public instruction tuning datasets for mathematical problem-solving: MetaMath (Yu et al., 2024) with 395K examples, MMIQC (Liu et al., 2024a) with 2.3 million examples, as well as vanilla rejection tuning (VRT) with 590K examples. Both DART-Math (Uniform) and DART-Math (Prop2Diff) use 590K training examples. **Right:** Number of responses for each query descending by difficulty across 3 synthesis strategies. Queries are from the MATH training split (Hendrycks et al., 2021). VRT is the baseline biased towards easy queries, while Uniform and Prop2Diff are proposed in this work to balance and bias towards difficult queries respectively. Points are

# Key Factors

-> SOTA synthetic datasets suffer from severe bias towards easy queries, and low coverage towards hard ones this is generally the case since, hard problems require more prompts than the easy ones to get them solved hence, reducing the ratio of hard to easy ones(typically due to same number sampling)

-> DART, a method that prioritizes more sampling trials for challenging queries, thereby generating synthetic datasets enriched with more responses for difficult questions compared to previous methods, 2 -forms:

Uniform which collects the same number of correct responses for all queries

Prop2Diff which biases the data samples towards the difficult queries, contrasting with vanilla rejection tuning

# Difficulty Analysis

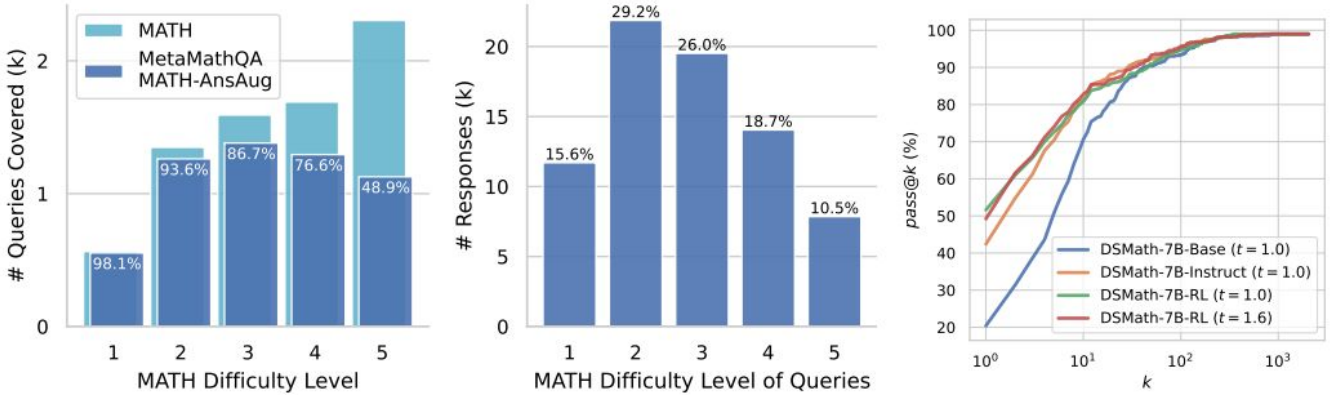



Figure 2: **Left:** Number of queries in the MATH training set and the MetaMathQA-MATH-AnsAug set across 5 difficulty levels annotated by humans. MetaMathQA-MATH-AnsAug is generated through rejection sampling from the original training queries. We annotate the query coverage ratio of MetaMathQA. While the most difficult queries (Level 5) are predominant in the original set, synthetic examples bias towards easier queries, dropping over 50% of the most difficult queries. **Middle:** Total number of responses for queries across different difficulty levels in MetaMathQA-MATH-AnsAug. The most difficult queries represent the smallest proportion, only accounting for 10.5% of all the samples. **Right:** $pass@k$ accuracy of different DeepSeekMath (DSMath) models and temperatures ($t$) on MATH500 (Lightman et al., 2024), a subset of MATH test set. With enough trials, models are actually able to sample out answer-correct responses to most (>99%) queries.

# Rejection Synthesis

-> Standard Instruction Tuning Consists of (x,y) pairs: x: problem, y: answer

-> For getting M responses, the notation is $\{(x_i, y_i^{(j)})\}_{j=1}^{M}$

-> For mathematical reasoning, usually filters are put in place to reject incorrect samples y(i) (based on whether the final answer in the synthetic response aligns with the ground-truth answer)

-> This method of filter is called rejection sampling and the tuning done sequentially is rejection tuning.

# Tuning Process

-> DART checks the capabilities of DeepSeekCoder2 7B, Meta Llama 7B, Mixtral 7B, and Meta Llama 70B

-> Uniform, which involves sampling responses for each query until each query accumulates ku correct responses, and ku is a preset hyperparameter determined by the desired size of the synthetic dataset

-> Prop2Diff, where we continue sampling responses until the number of correct responses for each query is proportional to its difficulty score. The most challenging queries will receive kp responses and kp is a hyperparameter

Eval Param: fail rate – the proportion of incorrect responses when sampling nd responses for a given query – as a proxy for difficulty

# Evaluation

In Domain:

GSM-8K and MATH

Out of Domain:

CollegeMath - ABout 2800 problems from 9 books in 7 fields

DeepMind-Mathematics - About 1000 problems

OlympiadBench-Math - About 700 math olympiad problems

TheormQA - About 800 problems on Math theorms

# Some Results

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| **70B General Base Model** | | | | | | | |
| Llama2-70B-Xwin-Math-V1.1† | 1.4M | 52.5 | 90.2 | 33.1 | 58.0 | 16.3 | 14.9 |
| Llama3-70B-ICL | – | 44.0 | 80.1 | 33.5 | 51.7 | 10.8 | 27.0 |
| Llama3-70B-MetaMath | 0.40M | 44.9 | 88.0 | 31.9 | 53.2 | 11.6 | 21.9 |
| Llama3-70B-MMIQC | 2.3M | 49.4 | 89.3 | 37.6 | 60.4 | 15.3 | 23.5 |
| Llama3-70B-VRT | 0.59M | 53.1 | 90.3 | 36.8 | 62.8 | 19.3 | **28.6** |
| DART-Math-Llama3-70B (Uniform) | 0.59M | 54.9 ↑1.8 | **90.4** ↑0.1 | **38.5** ↑1.7 | **64.1** ↑1.3 | 19.1 ↓0.2 | 27.4 ↓1.2 |
| DART-Math-Llama3-70B (Prop2Diff) | 0.59M | **56.1** ↑3.0 | 89.6 ↓0.7 | 37.9 ↑1.1 | **64.1** ↑1.3 | **20.0** ↑0.7 | 28.2 ↓0.4 |
| **7B Math-Specialized Base Model** | | | | | | | |
| DeepSeekMath-7B-ICL | – | 35.5 | 64.2 | 34.7 | 45.2 | 9.3 | 23.5 |
| DeepSeekMath-7B-Instruct | 0.78M | 46.9 | 82.7 | 37.1 | 52.2 | 14.2 | 28.1 |
| DeepSeekMath-7B-MMIQC | 2.3M | 45.3 | 79.0 | 35.3 | 52.9 | 13.0 | 23.4 |
| DeepSeekMath-7B-KPMath-Plus | 1.6M | 48.8 | 83.9 | – | – | – | – |
| DeepSeekMath-7B-VRT | 0.59M | 53.0 | **88.2** | **41.9** | 60.2 | 19.1 | 27.2 |
| DART-Math-DSMath-7B (Uniform) | 0.59M | 52.9 ↓0.1 | **88.2** | 40.1 ↓1.8 | 60.2 | 21.3 ↑2.2 | **32.5** ↑5.3 |
| DART-Math-DSMath-7B (Prop2Diff) | 0.59M | **53.6** ↑0.6 | 86.8 ↓1.4 | 40.7 ↓1.2 | **61.6** ↑1.4 | **21.7** ↑2.6 | 32.2 ↑5.0 |

# Questions To Ponder

-> Isn’t there a way to generalise the problem and then, accordingly work on it rather than just trying until the problems acheives some epoch or ratio?

# Improving Physics Reasoning in Large Language Models Using Mixture of Refinement Agents

By Raj Jaiswal

# Overview

-> LLMs have a hard time in reasoning not just mathematical but conceptual as well.

-> In terms of problem solving for physics issues faced are problem miscomprehension, incorrect concept application and mathematical inconsistency

-> Aim is to bridge the gap between proprietary and open source LLMs by using GPT-4 as an error identifier and guider.

-> Evaluated on  SciEval and MMLU subsets along with their own physics dataset (PhysicsQA).

# Related Works

-> LLMs for Reasoning: CoT, Auto-CoT, Iter-CoT and ToT

-> LLMs for Scientific Reasoning: LLMs don’t capture specific scientific knowledge

-> Self Verification: LLMs should be evaluate themselves on their own

-> Mathematical Reasoning: To do away with error in mathematical calculations

-> External Knowledge Base: To use KGs to impart outer world knowledge to the LLM for better understanding

# Problems at Hand

-> Miscomprehension:  LLMs in few cases struggle to fully grasp the objective of the question, along with misinterpreting the values of variables and constants provided in the question

-> Incorrect Application: LLMs struggle to apply the correct concepts or formulae with respect to the context of the given problem.

-> Mathematical Errors: Many physics problems involve mathematical reasoning and algebraic computation, areas where LLMs tend to struggle

# Handling the Errors at Hand

-> Problem Comprehension:

a) Object Alignment Flag - verifies whether the solution is focused on solving the correct objective of the given question
b) Variable Correctness Flag- verifies whether the solution uses the correct values for all variables and constants provided in the question, ensuring their correct values are applied in formulae and reasoning

-> Computation Verification Score:

$$Score_{\text{concept}} = \begin{cases} \frac{n}{N} & \text{if } 1 \leq n < N \text{ (error at step } n\text{)} \\ \frac{n}{N+1} & \text{if } n = N \text{ (error at last step)} \\ 1 & \text{if no errors occur} \end{cases}$$

# Refinement Agents

-> Miscomprehension Refinement: Prompt GPT for verification

-> Concept Refinement: Utilize an external physics knowledge base,

a) Error Identification & Thought Generation: LLM compare solution with KB
b) Concept Retrieval & Solution Refinement: The point at which the error occurs a retrieval method is used to improve the solution

-> Computational Refinement:

a) Code Generation: Creating a code to verify LLMs results
b) Correction: Using the generated code for refinement

# Initial Results

| Model | SciEval-Static | | | PhysicsQA | | | MMLU - High | | | MMLU - College | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AO | CoT | 3-Shot | AO | CoT | 3-Shot | AO | CoT | 3-Shot | AO | CoT | 3-Shot |
| LLaMa-3-70B | 70.07% | 82.23% | 63.41% | 38.37% | 56.76% | 59.29% | 60.16% | 72.88% | 73.66% | 59.41% | 71.76% | 71.76% |
| LLaMa 3.1 405B | **79.87%** | 89.63% | **82.92%** | **50.81%** | 76.75% | **78.37%** | **72%** | 91.52% | **88.98%** | **75.29%** | **88.23%** | **85.29%** |
| Gemma-2-27B | 60.36% | 79.26% | 53.04% | 39.18% | 54.59% | 59.45% | 55.93% | 77.11% | 74.45% | 51.11% | 73.52% | 67.64% |
| Gemini 1.5 Flash | 68.29% | 85.97% | 81.70% | 44.86% | 62.97% | 69.72% | 58.47% | 79.66% | 80.05% | 60.58% | 72.35% | 72.94% |
| GPT 3.5 Turbo | 41.46% | 66.46% | 48.78% | 28.10% | 42.70% | 42.70% | 47.45% | 58.47% | 33.89% | 35.29% | 50.58% | 42.35% |
| GPT4o | 64.02% | **92.68%** | 81.09% | 49.45% | **79.45%** | **78.37%** | 62.71% | **94.06%** | 87.28% | 70% | 84.70% | 84.17% |

# Model Results

| Model | Dataset | AO | COT | 3-Shot | MORA |
|---|---|---|---|---|---|
| **Gemma 2 27B** | MMLU College | 51.11% | 73.52% | 67.64% | **82.20%** |
| | MMLU High School | 55.93% | 77.11% | 74.45% | **75.88%** |
| | PhysicsQA | 39.18% | 54.59% | 59.45% | **70.62%** |
| | SciEval-Static | 60.36% | 79.26% | 53.04% | **88.76%** |
| **LLaMa 3 70B** | MMLU College | 59.41% | 71.76% | 71.76% | **78.82%** |
| | MMLU High School | 60.16% | 72.88% | 73.66% | **78.81%** |
| | PhysicsQA | 38.37% | 56.76% | 59.29% | **70.14%** |
| | SciEval-Static | 70.07% | 82.23% | 63.41% | **86.58%** |

# MoRA results

| Model | Dataset | AO | COT | 3-Shot | MORA |
|---|---|---|---|---|---|
| **Gemma 2 27B** | MMLU College | 51.11% | 73.52% | 67.64% | **82.20%** |
| | MMLU High School | 55.93% | 77.11% | 74.45% | **75.88%** |
| | PhysicsQA | 39.18% | 54.59% | 59.45% | **70.62%** |
| | SciEval-Static | 60.36% | 79.26% | 53.04% | **88.76%** |
| **LLaMa 3 70B** | MMLU College | 59.41% | 71.76% | 71.76% | **78.82%** |
| | MMLU High School | 60.16% | 72.88% | 73.66% | **78.81%** |
| | PhysicsQA | 38.37% | 56.76% | 59.29% | **70.14%** |
| | SciEval-Static | 70.07% | 82.23% | 63.41% | **86.58%** |

Table 3: Comparison of baseline approaches with MoRA across four datasets: SciEval-Static, PhysicsQA, MMLU High School and College based on final answer accuracy.

# Questions To Ponder

# MAGDI: Structured Distillation of Multi-Agent Interaction Graphs Improves Reasoning in Smaller Language Model

By Justin

# Introduction

-> MAGDI(Multi-Agent Interaction Graphs Distillation) use Multi Agent Graphs (MAGs) which help improve reasoning in SLM(Small Language Models), Language Models with about 3-4 billion parameters.

-> Benefit: As they are small in size computational costs required are less and it becomes more feasible to train SLM for their own downstream tasks rather than aiming for generalization.

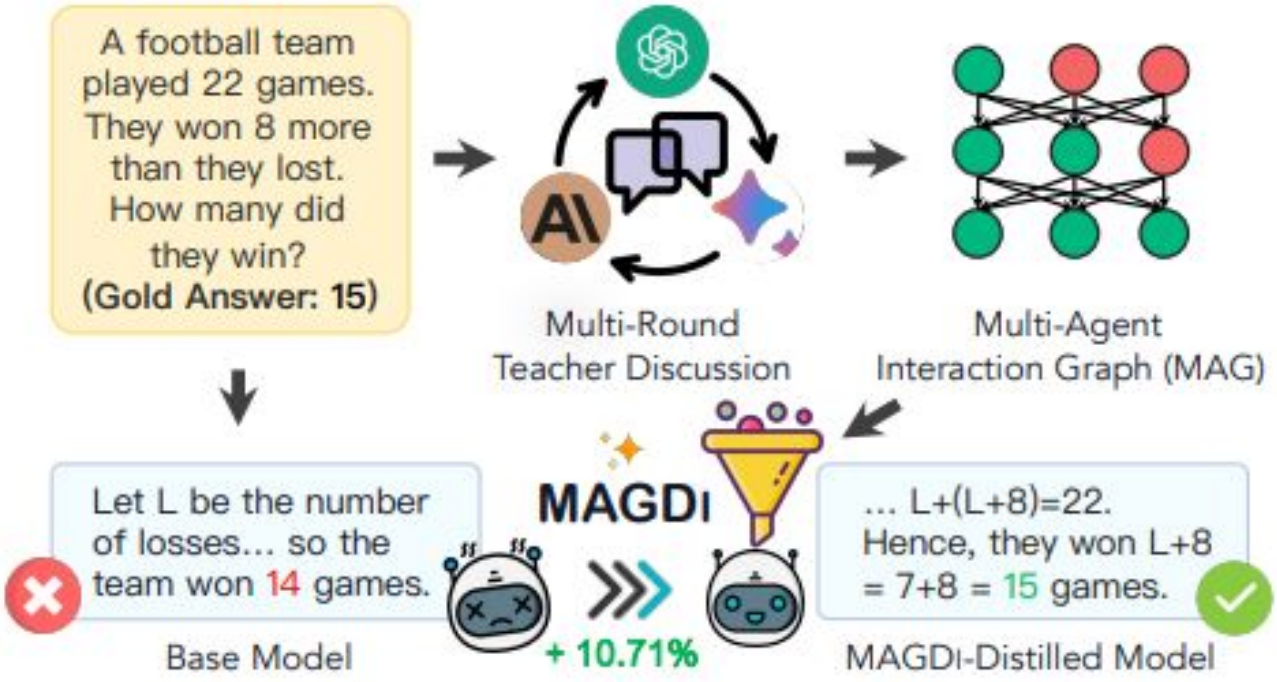
-> MAGs help to make better and valid choices

# Related Work

->Multi-Agent Reasoning: Large LLMs, like GPT-4 and Claude, have demonstrated improvements in reasoning tasks via interactive dialogues. However, these frameworks involve multiple rounds of discussion, making them computationally expensive and impractical for efficient deployment. Knowledge ->Distillation: Traditional methods transfer knowledge from a single large teacher model to a smaller student model. Multi-teacher approaches exist but fail to capture the interaction structures between teachers.

->Graph Modeling: Previous work used graphs to model human-human or human-model interactions, often for tasks like sentiment analysis or summarization. MAGDI uniquely focuses on LLM-to-LLM interactions to enhance reasoning in student models.

# Model Architecture



# Improvements Made

-> Multi-Agent Interaction Graphs (MAGs):

a) Represent reasoning interactions between LLMs as directed acyclic graphs.

b) Nodes represent reasoning chains, labeled as correct or incorrect, while edges capture discussion dependencies.

-> MAGDi

Uses GNNs to query over MAGS and use their respective information using constrastive loss and cross entropy loss

# Results

| Distillation Type | Distillation Data | Distilled Model | Datasets | | | | | Average Acc |
|---|---|---|---|---|---|---|---|---|
| | | | StrategyQA | CSQA | ARC-c | GSM8K | MATH | |
| No Teacher (Jiang et al., 2023) | - | Mistral-7B-Instruct | 61.57 | 57.89 | 60.32 | 44.05 | 7.02 | 46.17 |
| Single-Teacher (Li et al., 2023; Magister et al., 2023; Fu et al., 2023; Ho et al., 2023) | Claude2 | SIT-Claude2 | 64.39 | 64.18 | 68.24 | 45.34 | 7.24 | 49.89 |
| | Bard | SIT-Bard | 68.56 | 65.06 | 66.87 | 45.61 | 7.06 | 50.63 |
| | GPT-4 | SIT-GPT4 | 69.96 | 66.87 | 68.91 | 47.38 | 8.24 | 52.27 |
| Multi-Teacher | All Nodes | DSS-MT (Hsieh et al., 2023) | 71.18 | 69.42 | 71.38 | 51.84 | 9.98 | 54.76 [+ 2.49%] |
| | Round-0 Nodes | MAGDI-R0 [Level 1] | 71.18 | 67.36 | 72.06 | 48.52 | 9.72 | 53.77 [+ 1.50%] |
| | Correct Nodes | MAGDI-CN [Level 2] | 71.62 | 69.31 | 72.34 | 50.11 | 10.66 | 54.81 [+ 2.54%] |
| | All Nodes | MAGDI-AN [Level 3] | 72.10 | 70.65 | 71.92 | 50.69 | 11.98 | 55.47 [+ 3.20%] |
| | MAG | MAGDI [Level 4] | **74.24** | **72.56** | **72.61** | **52.27** | **12.76** | **56.88 [+ 4.61%]** |

# Questions To Ponder

# Understanding the Reasoning Ability of Language Models From the Perspective of Reasoning Paths Aggregation

By Xinyi et al.

# Introduction

->The study aims to understand how reasoning abilities emerge in language models during pre-training with a next-token prediction objective.

->It hypothesizes that reasoning arises from aggregating indirect reasoning paths encountered during pre-training.

->Focuses on two reasoning scenarios:

a) Logical Reasoning with Knowledge Graphs (KGs).
b) Chain-of-Thought (CoT) reasoning.

# Idea

-> Language models aggregate probabilities of reasoning paths seen during pre-training.

-> Logical Reasoning (KG Reasoning):

a) Pre-trained models on random walk paths from KGs.
b) Evaluated their ability to predict missing links in KGs.

-> Mathematical Reasoning (CoT Reasoning):

a) Created random walk reasoning paths from existing CoT training data.
b) Extended training with these paths to test improvements in reasoning tasks.