End-to-End Bangla AI for Solving Math Olympiad Problem Benchmark:Leveraging Large Language Model Using Integrated Approach

H.M.Shadman Tabib and Jaber Ahmed Deedar

Department of Computer Science and Engineering,Bangladesh University of Engineering and Technology
Dhaka,Bangladesh

Abstract. This work introduces systematic approach for enhancing large language models (LLMs) to address Bangla AI mathematical challenges. Through the assessment of diverse LLM configurations, fine-tuning with specific datasets, and the implementation of Retrieval-Augmented Generation (RAG), we enhanced the model's reasoning precision in a multilingual setting. Crucial discoveries indicate that customized prompting, dataset augmentation, and iterative reasoning improve the model's efficiency regarding Olympiad-level mathematical challenges.

Keywords: Large Language Models (LLMs), Fine-Tuning, Bangla AI, Mathematical Reasoning, Retrieval-Augmented Generation (RAG), Multilingual Setting, Customized Prompting, Dataset Augmentation, Iterative Reasoning, Olympiad-Level Challenges

1 Introduction

In recent years, large language models (LLMs) have exhibited tremendous potential in tackling hard mathematics problems, including ones at Olympiad-level complexity. Omni-math is a global benchmark that covers several sub-domains and difficulty levels[1]. Odyssey math data measures mathematical problem-solving ability in LLMs[2]. These are just a few of the tools that researchers have made to test and improve model reasoning. Researchers have devised numerous methodologies and standards to evaluate and enhance the effectiveness of these models in mathematical reasoning.

A significant method is the self-critique pipeline, implemented in the ChatGLM-Math model. This approach entails training a Math-Critique model based on the LLM to provide feedback signals. The LLM's own outputs then undergo rejective fine-tuning and direct preference optimization for data acquisition. Experiments with the ChatGLM3-32B model showed that this pipeline greatly enhances the ability to solve mathematical problems while maintaining language skills, making it better than larger LLMs. [3]

Another new technique involves the creation of MetaMath, a refined language model focused on mathematical reasoning. The MetaMath methodology initiates by formulating mathematical inquiries from several viewpoints, culminating in a novel dataset termed MetaMathQA. This dataset used to fine-tune the LLaMA-2 models. Studies using benchmarks like GSM8K and MATH show that MetaMath performs better than a number of open-source LLMs. For example, the MetaMath-7B model achieved a GSM8K precision of 66.5% and a MATH accuracy of 19.8%, which is much better than the performance of models of the same size. [4]

New standards have been established to assess the mathematical reasoning capabilities of LLMs, in addition to model-specific improvements. Mathador-LM is a dynamic benchmark derived from the Mathador game, wherein the goal is to get a target number

DIO:10.5121/ijnlc.2024.13604

through fundamental arithmetic operations applied to a specified set of base numbers, adhering to straightforward criteria. This test integrates ruleset interpretation, planning, and problem-solving, offering a thorough evaluation of LLMs' mathematical reasoning ability. Research indicates that modern models do poorly on Mathador-LM, achieving scores much below those of average third graders, underscoring the necessity for enhancements in this domain.[5]

Additionally, ConceptMath is a multilingual benchmark (English and Chinese) that evaluates concept-specific mathematical reasoning in LLMs. ConceptMath organizes math problems in a way that lets them be tested at different levels of detail with concept-specific accuracy. This is different from traditional benchmarks that only measure general mathematical reasoning with average accuracy. Evaluations have shown that existing LLMs, while achieving high average accuracies on common metrics, exhibit significant performance variations among different mathematical ideas and may even struggle with basic ones.[6]

Another paper [7] introduces the MCT Self-Refine (MCTSr) algorithm, a novel approach integrating Large Language Models (LLMs) with Monte Carlo Tree Search (MCTS) to improve performance in complex mathematical reasoning tasks. The MCTSr algorithm addresses the challenges of accuracy and reliability in LLM-based reasoning by employing systematic exploration and heuristic self-refinement mechanisms. Another work presents OlympiadBench, a bilingual multimodal benchmark consisting of 8,476 Olympiad-level mathematics and physics tasks, intended to assess the advanced reasoning abilities of Large Language Models (LLMs) and Large Multimodal Models (LMMs). Preliminary assessments indicate considerable difficulties, with the top-performing model, GPT-4V, attaining merely 17.97% accuracy, highlighting the benchmark's stringency and its potential as a tool for furthering AGI research.[8]

These advancements highlight the continuous endeavors to improve the mathematical problem-solving capabilities of LLMs via novel training methodologies and thorough assessment standards.Still, there are difficulties making sure LLMs can operate across several languages and problem types—especially for challenging tasks in non-English environments. Emphasizing the continuous improvement of these models, efforts including multilingual machine translation[9] and optimized frameworks like BEATS, which strengthens mathematical reasoning through back-verification and adaptive disambiguation[10], show this study builds on previous research by concentrating on fine-tuning LLMs to solve Bangla math problems. It does this by using a structured strategy of dataset enhancement, retrieval-augmented generation (RAG) [16], and personalized prompting to get the best results for Bangla AI math problems.

2 Methodology

2.1 Model Selection

In our AI-driven Bangla Math Olympiad solution, we tested many large language models (LLMs) to find the best one for mathematical reasoning in Bangla. Our selection process evaluated models based on their mathematical problem-solving, Bangla flexibility, and computing economy.

Specifically designed for mathematical problem-solving, **Deepseek-Math-7b-Instruct** [11]was evaluated. It did not perform well in our used dataset.

Next, we evaluated the **NuminaMath** [12] models, which include **NuminaMath-7B-TIR** and **NuminaMath-7B-CoT** .These models were used to solve the problems in

AI Mathematical Olympiad organized by AIMO. This model performs moderately well on our used dataset, but not up to expected level due to language difference.

We then focused on the **Qwen2.5-7B-Instruct, Coder, and Math Instruct** models[13]. Our Bangla evaluations of the Qwen2.5-7B-Math Instruct model for mathematical exercises were not enough satisfactory. Qwen2.5-7B-coder, which is transcribed for coding, also failed to generate codes for most Bangla languages. Manual inference for some selected problems revealed that they could not solve trivial problems.

However, Qwen2.5-7B-Instruct understood and solved Bangla maths problems well. The model's versatility and Bangla language adaptability made it excellent for our purpose. It balanced mathematical thinking with Bangla language adaptation better for our purposes after extensive evaluations. Next, we tested on Qwen2.5-32B-Instruct-AWQ without fine-tuning or RAG, which makes a more improved accuracy over test set marking 77 out of 100 showing best of all. It implies that using model trained with large parameter ensures better performance.

2.2 Datasets

- **BDMO Dataset**
  This dataset is provided by Bangladesh Math Olympiad (BDMO). This dataset originally contains 209 Bangla Math Olympiad problems with their numerical solutions. We extended the dataset by adding step-by-step TIR solutions. We also translated the problem statements into English for further testing. There are also 100 test problems available in this kaggle contest **here** upon which the accuracy score is calculated.
- **Translated Numina CoT Dataset:**
  This dataset comprises 0.8M problems formatted using the Chain of Thought reasoning technique, focusing on step-by-step solutions which were used to train NuminaMath-7B-CoT in AI-MO Competition. For our use case, these were translated using gemini-1.5-flash model into Bangla for fine-tuning. It supports the training of models capable of multi-step reasoning. This contains both problem statement and problem solution step description in Bangla.

- **Translated Numina TIR Dataset:**
  A foundational dataset with around 72K math competition problems ,with each solution generated by GPT-4 with tool integrated reasoning (TIR)[15]. Next, these were also translated likewise for our convenience.

- **Synthetic Problems Dataset:**
  This dataset includes a variety of synthetic mathematical problems generated using the GPT4o API based on BDMO dataset. These problems ensure multiple similar problems with paraphrasing in order to ensure diversity while training for each category.

These datasets can be found in the Dataset Availability section.

2.3 Preprocessing

We underwent a comprehensive data preprocessing method that integrated Retrieval-Augmented Generation (RAG) with keyword search to solve the challenges associated with Bangla Math Olympiad. We initially identified significant keywords from each mathematical issue manually, focusing on terms that focus on the essential concepts and categories of the problems. Utilizing these keywords, we conducted similarity searches inside our problem dataset to identify issues with analogous structures or themes. The recognized similar problems and their solutions as Tool Integrated Reasoning were then incorporated as few-shot instances in the model's prompts, providing contextual assistance and enhancing the model's problem-solving capabilities. By integrating RAG with keyword-based retrieval, we enhanced the model's ability to comprehend and address Bangla Math Olympiad problems effectively. This approach associated contextual examples with each problem, assists in model's effectiveness in solving Math problems.

2.4 Augmentation

We expanded and diversified our training dataset in order to enhance the capabilities of our model with the utilization of data augmentation techniques, to solve the challenges presented by the Bangla Math Olympiad. With the help of OpenAI's GPT-4.0, we generated five paraphrased and similar versions for each of the original problems. These ensure that the original difficulties' complexities and nuances were maintained. After the implementation of this modification, heterogeneity was introduced into the dataset, which involve manual categorization over the problem sets. By exposing the model to a greater variety of problem statements, we were able to improve its flexibility and robustness through the process of addressing a variety of mathematical challenges in Bangla .

2.5 Fine-tuning

Fine-tuning of the Qwen-2.5-7B-Instruct model was done using **1x NVIDIA H100 NVL GPU** for efficient training. The model underwent three phases to improve its reasoning for Bangla AI mathematical problems and solutions. The first phase involved fine-tuning the model using the translated Numina TIR dataset, which had around 72,000 problems and step-by-step python solutions. The second stage involved using the Numina CoT dataset, which included over around 800,000 CoT problem solution pairs. For our use case, we used 250,000 of those. Then, we used GPT4o API for creating synthetic datasets to increase the model's adaptability and versatility. The final stage focused on paraphrasing and augmentation, rephrasing problem sets to simulate different mathematical presentations. The model was evaluated using BDMO dataset and test dataset at each level to examine accuracy improvements and fine-tuning stage effects. Several rounds of model training improved the model's ability to understand complex reasoning processes and improve its Bangla AI mathematics comprehension and response formulation. All the models fine-tuned in each step can be found in the Model Availability section.

2.6 Model Architecture and Flow

Figure 1 illustrates the architecture of our system developed to analyze the Bangla mathematical problem. The system employs a Large Language Model (LLM) in conjunction with TIR agents and self-consistency [14]. The TIR agents generate python code iteratively until the correct solution is found and then executes them in REPL (Read Eval Print Loop). According to self-consistency prompting, final answer is chosen by majority voting among the differently generated solutions.

The procedure starts by accepting Bangla mathematics questions as input. Keywords are derived analyzing the training data and a Retrieval-Augmented Generation (RAG) technique is employed to identify analogous instances.They are basically categorized as similar problems matching their keywords and then used these similar problems for few

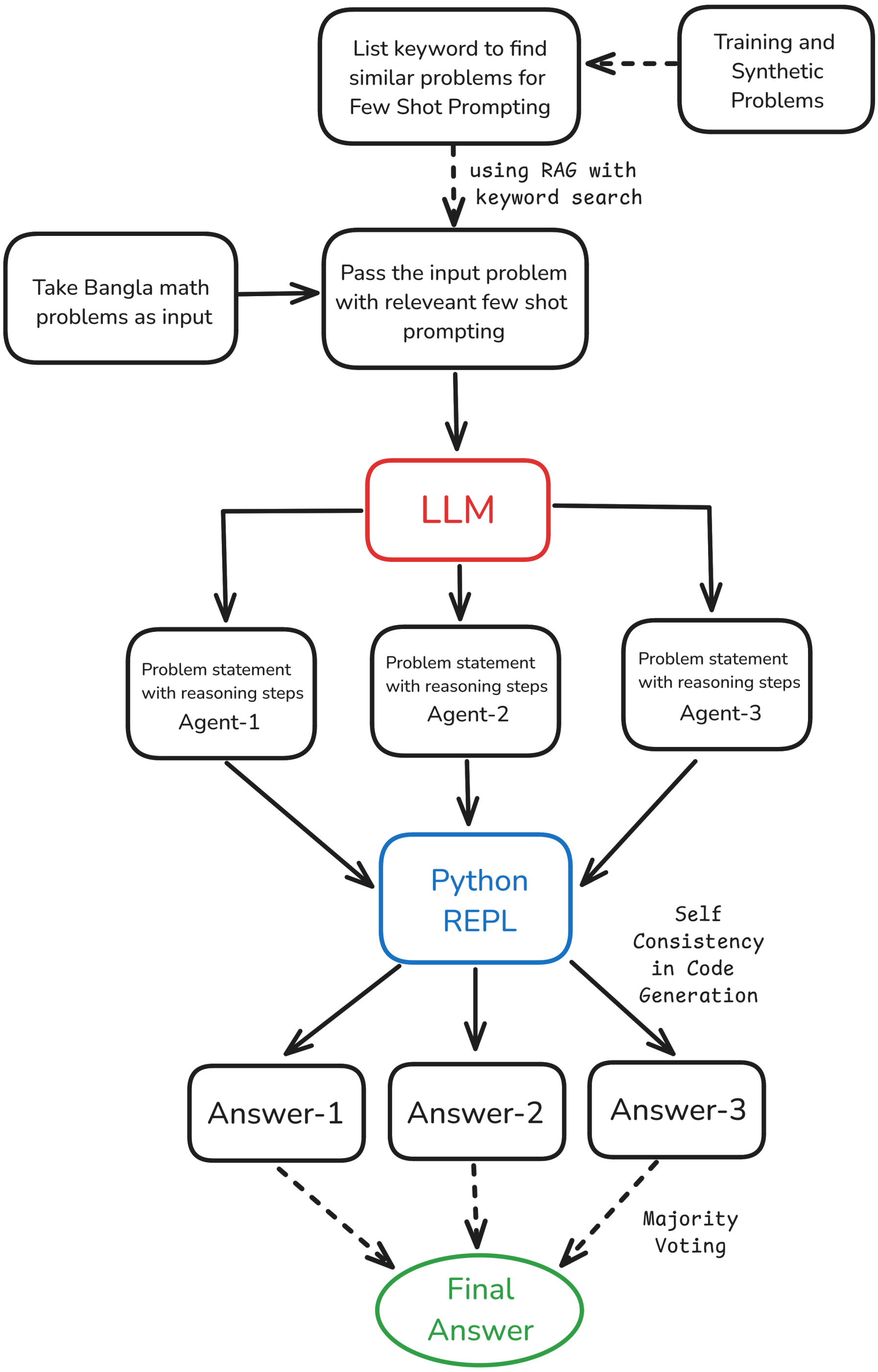
List keyword to find similar problems for Few Shot Prompting
Training and Synthetic Problems
using RAG with keyword search
Take Bangla math problems as input
Pass the input problem with releveant few shot prompting
LLM
Problem statement with reasoning steps Agent-1
Problem statement with reasoning steps Agent-2
Problem statement with reasoning steps Agent-3
Python REPL
Self Consistency in Code Generation
Answer-1
Answer-2
Answer-3
Majority Voting
Final Answer

Fig. 1. Model Architecture

shot prompts. These examples assist the LLM in producing methodical solutions. Multiple TIR agents develop distinct solution paths to investigate several methods for addressing the challenge, subsequently testing them in a Python code execution environment.

Upon the failure of a solution path, the system re-evaluates the problem to rectify any inaccuracies. Ultimately, the system aggregates the agents' results, employing a voting mechanism to choose the optimal option. This methodology guarantees precision and dependability through the integration of reasoning, verification, and consensus techniques.

2.7 Model Inference and Prompting

The TIR agents were tested using system prompts containing no instruction and prompts containing tailored instructions. These instructions were based on the problems' categories which were manually curated investigating the training dataset. Such of these can be found here. Moreover, we tested both zero shot and few shot prompting techniques. For few shot examples, we used keyword search RAG where problems are categorized matching relevant keywords.

Prompts Used

Base Prompt

```
Here is a math problem in Bengali:
{problem}

The answer is a non-negative integer. Please reason step by step to solve the problem
    above. Provide Python code to verify your reasoning.

Put your final integer answer within \boxed{}.
```

Tailored Instruction for Number Theory Problems

```
Please write brute force solution codes to answer.
```

Translation Prompt

```
Please translate the problem to English first. Then solve the problem.
```

Advanced Prompt

```
Solve the problem in Bengali by explaining all intermediate steps in English.
Use Python to verify the solution.
Highlight important mathematical reasoning.
```

Step-by-Step Guidance Prompt

```
Explain the problem-solving process using detailed reasoning.
Start from basics and gradually solve the problem step by step. Show intermediate results
clearly.
Verify with Python.
```

Model Configuration	**Temperature**	**Problem Language**	**Reasoning Language**	**Result**
GPT4o - basic	0.7	Bangla	English	115 / 209
GPT4o - basic	0	Bangla	English	**122*** / 209
GPT4o - basic	0.7	Bangla	Bangla	109 / 209
GPT4o - basic	0	Bangla	Bangla	107 / 209
GPT4o - TIR	0	Bangla	English	125 / 209
GPT4o - TIR	0	English	English	**130**** / 209

Table 1. Comparison of GPT-4o Configurations for BDMO dataset problems. Here ”basic” means direct prompting without using any tool and ”TIR” denotes model is coupled with tool integrated reasoning.

Model Configuration	**Problem Language**	**Reasoning Language**	**Result**
GPT4o - Basic	Bangla	Bangla	50 / 100
GPT4o - Basic	English	Bangla	62 / 100
GPT4o - Basic	Bangla	English	**64*** / 100
GPT4o - Basic	English	English	60 / 100
GPT4o - TIR	Bangla	Bangla	63 / 100
GPT4o - TIR	English	Bangla	61 / 100
GPT4o - TIR	Bangla	English	67 / 100
GPT4o - TIR	English	English	**71**** / 100

Table 2. Comparison of GPT-4o Configurations for test problems. Here ”basic” means direct prompting without using any tool and ”TIR” denotes model is coupled with tool integrated reasoning.

Model Configuration	**Problem Language**	**Reasoning Language**	**Result**
Deepseek-Math-7B-Instruct-Basic	Bangla	Bangla	28 / 100

Table 3. Comparison of DeepSeekMath model for test problems. Here ”basic” means direct prompting without using any tool.

Count of TIR Agents	**Instruction Given**	**Language**	**Result**
42	NO	Bengali	59 / 100
10	NO	English	62 / 100
13	NO	English	53 / 100
10	YES	English	**64*** / 100
5	YES	English	62 / 100
5	YES	Bengali	50 / 100

Table 4. Benchmark Results for Numina-7B-TIR Model. Here, ”no instructions” implies no system prompt was used during inference, while ”instructions given” denotes tailored system prompts.

Model Configuration	**Problem Language**	**Samples**	**Depth**	**Score**
Qwen2.5-7B-Math-Instruct	English	10	10	59
Qwen2.5-7B-Math-Instruct	English	5	10	57
Qwen2.5-7B-Coder Instruct	English	5	-	65
Qwen2.5-7B-Instruct	English	5	5	70
Qwen2.5-7B-Instruct	English	50	9	70
Qwen2.5-7B-Instruct	Bangla	50	9	68
Qwen2.5-7B-Instruct (With prompt to translate in English first)	Bangla	50	9	64
Qwen2.5-32B-Instruct-AWQ	Bangla	10	4	**77***

Table 5. Performance Comparison of Qwen2.5 Family Models for Solving Bangla AI Math Problems

Dataset	**Data Size**	**Epochs**	**Augmentation**	**Score**
TIR	72K	1	No	70
CoT	0.25M	1	No	40
TIR	72K TIR	3	No	70
CoT + TIR	72K TIR+ 0.25M CoT	5	Yes	68
CoT + TIR + RAG	72K TIR+ 0.25M CoT	5	Yes	**71***
TIR + RAG	72K TIR	5	Yes	**71***

Table 6. Performance Comparison of Fine-Tuned Models of Qwen2.5-7B-Instruct with Different Datasets for Solving Bangla AI Math Problems

3 Results

The study aimed to improve Bangla AI mathematics performance by finding the optimum model configuration, tweaking it, and applying RAG. The initial stage involved testing GPT-4o for reasoning and linguistic proficiency. GPT4o-TIR had the highest accuracy of 130 out of 209 in BDMO dataset. When Bangla was the main language, the model performed worse, suggesting the need for more advanced and versatile model. The second stage examined how number of TIR agents and instructional setups affected the Numina-7B-TIR model's performance. The maximum score was 64 out of 100 (here we used the test dataset) with 10 agents per problem and clear English direction as prompt. Next, we tested Qwen2.5-7B-Instruct. It was more versatile and effective arrangement, earning 70 with 5 agents and 5 depth. In the final testing phase, data augmentation with TIR, CoT, and RAG datasets enhanced model accuracy a little bit,increased up to 71. Next at the end, a larger variant of Qwen, named Qwen2.5-32B-Instruct-AWQ outperformed all other remaining models in solving problems signifying the fact that it improves with parameter enhancement. This one might be improved too with help of our additional fine-tuning,RAG and prompt techniques.The study concluded that retrieval-augmented methods and well-curated datasets improve model performance in multilingual, specialized domains.

4 Experiments and Discussion

In this section, we discuss the various experiments conducted to evaluate the model's performance in solving Bangla AI math problems. Each experiment focuses on specific aspects of problem-solving, including problem categorization, prompt structuring, multilingual reasoning, and response nuances. These insights provide valuable guidelines for optimizing the model's performance on different types of mathematical problems in Bangla. From **Table 1** and **Table 2**,it is further observed that if the problem statement is in Bangla and the solution steps are described in English, there is an enhanced accuracy score over the datasets while using direct prompting without any tool integration.

4.1 Problem Categorization for Improved Performance

We observed that categorizing problems into specific types improves the model's ability to generate accurate responses. When problems are grouped by category—such as Number Theory, Geometry, and Combinatorics—the model benefits from prompts tailored to each category's unique solving techniques.

4.2 Tailored Prompting for Problem Types

Specific types of problems showed better performance when prompted with targeted solving techniques:

- **Number Theory**: The model performs better on number theory problems if prompted to solve using brute force methods.
- **Geometry**: In some cases, geometry problems are more effectively solved when the prompt includes instructions to approach the solution with coordinate geometry transformations.
- **Combinatorics and Functional Equations**: Problems involving combinatorics, functional equations, and recursion show improved performance when prompted with hints about dynamic programming techniques.

4.3 Multilingual Reasoning and Repetitive Querying

In general, the model achieves higher accuracy in English than in Bangla. However, during experimentation, we found that if multiple queries are submitted—one translated into English and the other kept in its original Bangla form—the model can enhance its reasoning capabilities through repetitive reasoning. This approach allows the model to analyze the problem in different linguistic contexts, resulting in more accurate solutions.

4.4 Prompt Phrasing and Politeness

Interestingly, a few cases revealed that the model follows prompts more strictly if the word "please" is included at the beginning of the prompt. Although unconventional, this observation suggests that polite phrasing can influence the model's adherence to prompt instructions in certain scenarios.

5 Conclusion and Future Work

The overall integrated approach used by ourselves suggests that a proper fine tuning with augmented datasets provide comparatively a slight better result. The overall evaluation score would improve provided the model is fine tuned with Bangla Mathematics dataset. In our experiment. incorporating RAG which involves keyword search based similarity does not improve up to satisfactory level.Model performance may be better in case of using better retrieval models for RAG.Same goes for our large language model too. The more higher parameter a language model is, the more accurate result it would produce too. It is probable that a more curated and well organized dataset fine tuning may lead to more improvised and upgraded result in future.

Aspect	**Observation**	**Recommendation**
Problem Categorization	Categorizing problems into types (e.g., Number Theory, Geometry) improves accuracy	Group problems by type for tailored prompts
Number Theory Prompts	Model performs better with brute force prompts for number theory	Include brute force hints in prompts for Number Theory
Geometry Prompts	Coordinate geometry approaches improved model performance in some cases	Prompt with coordinate geometry conversion hints for Geometry
Combinatorics and Functional Equations	Dynamic programming prompts enhance responses for combinatorial, functional, and recursive problems	Use dynamic programming prompts for these problem types
Multilingual Reasoning	Model performs better in English than in Bangla; repetitive reasoning improves results	Use both Bangla and translated English queries for repetitive reasoning
Politeness in Prompts	Adding “please” to the prompt sometimes results in stricter adherence	Consider polite phrasing like “please” in prompts to increase adherence

Table 7. Observations and Recommendations for Enhancing AI Math Problem Solving

6 Dataset Availability

All the mentioned datasets can be downloaded from **here**.

7 Model Availability

All the fine-tuned model can be downloaded from **here**.

References

1. Gao, B., Song, F., Yang, Z., Cai, Z., Miao, Y., Dong, Q., ... & Chang, B. (2024). Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985.*
2. Fang, M., Wan, X., Lu, F., Xing, F., & Zou, K. (2024). Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *arXiv preprint arXiv:2406.18321.*
3. Xu, Y., Liu, X., Liu, X., Hou, Z., Li, Y., Zhang, X., ... & Dong, Y. (2024). ChatGLM-Math: Improving Math Problem-Solving in Large Language Models with a Self-Critique Pipeline. arXiv preprint arXiv:2404.02893. *arXiv preprint arXiv:2404.02893.*

4. Yu, L., Jiang, W., Shi, H., Yu, J., Liu, Z., Zhang, Y., ... & Liu, W. (2023). Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284.*
5. Kurtic, E., Moeini, A., & Alistarh, D. (2024). Mathador-LM: A Dynamic Benchmark for Mathematical Reasoning on Large Language Models. *arXiv preprint arXiv:2406.12572.*
6. Wu, Y., Liu, J., Bu, X., Liu, J., Zhou, Z., Zhang, Y., ... & Zheng, B. (2024). Conceptmath: A bilingual concept-wise benchmark for measuring mathematical reasoning of large language models. *arXiv preprint arXiv:2402.14660.*
7. Zhang, D., Huang, X., Zhou, D., Li, Y., & Ouyang, W. (2024). Accessing gpt-4 level mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b. *arXiv preprint arXiv:2406.07394.*
8. He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., ... & Sun, M. (2024). Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008.*
9. Zhu, W., Liu, H., Dong, Q., Xu, J., Huang, S., Kong, L., ... & Li, L. (2023). Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675.*
10. Sun, L., Liang, H., & Zhang, W. (2024). BEATS: Optimizing LLM Mathematical Capabilities with BackVerify and Adaptive Disambiguate based Efficient Tree Search. *arXiv preprint arXiv:2409.17972.*
11. Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., ... & Guo, D. (2024). Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300.*
12. Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S., ... & Polu, S. (2024). Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository, 13.*
13. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., ... & Qiu, Z. (2024). Qwen2. 5 Technical Report. *arXiv preprint arXiv:2412.15115.*
14. Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... & Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models.*arXiv preprint arXiv:2203.11171.*
15. Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Huang, M., ...& Chen, W. (2023). Tora: A tool-integrated reasoning agent for mathematical problem solving.. *arXiv preprint arXiv:2309.17452.*
16. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems, 33, 9459-9474.*