Multilingual Mathematical Reasoning: Advancing Open-Source LLMs in Hindi and English

Avinash Anand, Kritarth Prasad, Chhavi Kirtani, Ashwin R Nair, Manvendra Kumar Nema, Raj Jaiswal, Rajiv Ratn Shah

Indraprastha Institute of Information Technology, Delhi
{avinasha, kritarth20384, chhavi18229, ashwin20037, manvendra22038, Jaiswalp, rajivratn}@iiitd.ac.in

Abstract

Large Language Models (LLMs) excel in linguistic tasks but struggle with mathematical reasoning, particularly in non-English languages like Hindi. This research aims to enhance the mathematical reasoning skills of smaller, resource-efficient open-source LLMs in both Hindi and English. We evaluate models like OpenHathi 7B, LLaMA-2 7B, WizardMath 7B, Mistral 7B, LLeMMa 7B, MAmmoTH 7B, Gemini Pro, and GPT-4 using zero-shot, few-shot chain-of-thought (CoT) methods, and supervised fine-tuning. Our approach incorporates curriculum learning, progressively training models on increasingly difficult problems, a novel Decomposition Strategy to simplify complex arithmetic operations, and a Structured Solution Design that divides solutions into phases. Our experiments result in notable performance enhancements. WizardMath 7B exceeds Gemini's accuracy on English datasets by +6% and matches Gemini's performance on Hindi datasets. Adopting a bilingual approach that combines English and Hindi samples achieves results comparable to individual language models, demonstrating the capability to learn mathematical reasoning in both languages. This research highlights the potential for improving mathematical reasoning in open-source LLMs.

Code and Dataset — https://github.com/midas-research/Multilingual-Mathematical-Reasoning.git

Extended version — https://arxiv.org/abs/2412.18415

Introduction

Enhancing AI systems to solve complex problems has become a crucial objective within the AI research community, particularly in the realm of mathematical question answering. While models like GPT-4 and Gemini have demonstrated their strengths in arithmetic (Zhang et al. 2024), algebra(Kao, Wang, and Hsieh 2024), scientific text generation(Anand et al. 2024d, 2023a), and symbolic manipulation(Dave et al. 2024), they are not without limitations. Our evaluations on the GSM8K (Cobbe et al. 2021) and MATH (Hendrycks et al. 2021) datasets reveal a stark contrast in their capabilities. These models perform well on the relatively straightforward GSM8K dataset, but their effectiveness significantly diminishes when tasked with the more challenging MATH dataset. This dataset includes high-school competition-level questions that require a deeper level of contextual understanding and more advanced reasoning skills. The discrepancies in performance highlight the current limitations of these models in handling complex mathematical problem-solving.

In addition to these challenges, there is a noticeable gap in the performance of large language models (LLMs) when applied to English versus non-English languages, particularly in natural language processing tasks such as question answering and classification. This gap is particularly evident in Hindi, India's predominant language, which is used by over 105 million students according to UDISE+ reports for 2019-20[1]. Enhancing the capabilities of LLMs in Hindi is essential to make these tools more accessible and effective in subject-specific learning contexts. While research efforts such as OpenHathi-7B (AI 2023), Hi-NOLIN (Research 2023), and Airavata (Gala et al. 2024) have made strides in adapting LLMs to the Hindi language, these models were not originally optimized for domain-specific tasks like mathematical problem-solving.

Recent advancements in open-source LLMs have shown promise in improving mathematical and physics problem-solving abilities (Anand et al. 2024a,c,b, 2023b), as evidenced by prominent models like WizardMath (Luo et al. 2023), Mistral (Jiang et al. 2023), LLeMMA (Azerbayev et al. 2023), and MAmmoTH (Yue et al. 2023). However, these advancements have largely focused on the English language, with limited performance gains observed in Hindi math datasets. Additionally, closed-source models such as GPT-4 and Gemini-Pro continue to outperform open-source models on established benchmarks like GSM8K and MATH, as well as on newly defined Hindi datasets. The significant performance disparity can be attributed to the vast difference in the number of parameters these models are trained on. While the open-source LLMs explored in this research have fewer than 10 billion parameters, well-known closed-source LLMs are trained on considerably large parameter counts. Given the constraints on computational resources, this research focuses on enhancing the performance of smaller

[1]The Unified District Information System for Education Plus (UDISE+) report is a comprehensive database that provides detailed information about schools and educational statistics across India.

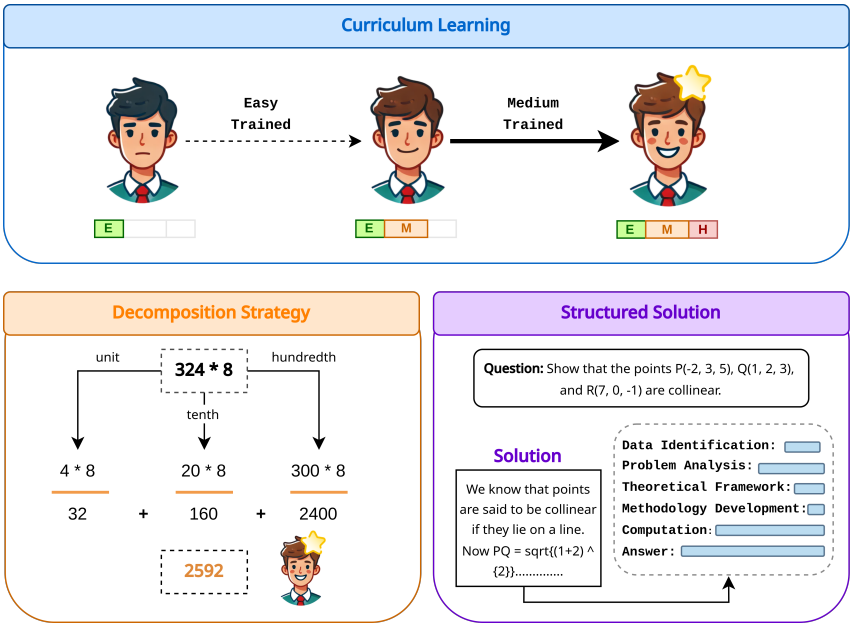

Figure 1: Curriculum Learning with Structured Solutions: A Comprehensive Framework to Gradually Guide Models Through Complex Mathematical Challenges.

open-source LLMs (SLLMs), acknowledging the limitations while seeking to optimize within these parameters.

This research introduces several key contributions aimed at improving the mathematical capabilities of SLLMs, particularly in Hindi:

1. **Introduction of the Decomposition Strategy:** A novel approach designed to enhance SLLMs' ability to solve complex mathematical operations by breaking them down into smaller, more manageable components in the enhanced HAWP dataset (see Figure 1, 2).
2. **Structured Solution Approach with Curriculum Learning:** A combined methodology that integrates a structured solution framework with Curriculum Learning, as illustrated in Figure 1, 2. This approach progressively guides models through increasingly complex mathematical problems, enhancing their problem-solving abilities.
3. **Bilingual Combined Training:** We propose the methodology of Bilingual Combined Training, where the Structured Solution Approach with Curriculum Learning is applied on a dataset containing both English and Hindi versions of Mathematical questions-answers.
4. **IndiMathQA Dataset Creation:** We developed the IndiMathQA dataset by curating 598 math problems from NCERT [2] textbooks for grades 10-12, spanning 14 mathematical domains. Expert annotations categorized these into easy, medium, and hard problems. The dataset was expanded to 7,823 questions.
5. **Performance Analysis of Multilingual LLMs:** A comprehensive analysis of several LLMs, including five open-source English LLMs, three open-source Hindi LLMs, and two closed-source models, was conducted using both English and Hindi mathematical datasets to assess their capabilities and limitations.

[2]NCERT is a major textbook for school students in India.

Related Work

Recent advances in large language models (LLMs) have significantly improved their ability to perform complex tasks, particularly in the areas of natural language processing and mathematical reasoning. However, one area that remains underexplored is the application of Curriculum Learning to these models. Originally proposed by Bengio et al. (Bengio et al. 2009), Curriculum Learning is a training strategy that mimics the way humans learn by gradually increasing the complexity of tasks presented to the model. Although widely used in deep learning, its application to LLMs has been limited, particularly in enhancing the models' capabilities in complex, multistep reasoning tasks (Soviany et al. 2022). This section discusses various open-source and bilingual LLMs, their architectures, and the benchmark datasets used to evaluate their performance.

Open-Source Large Language Models

Recent advancements in open-source large language models (LLMs) have demonstrated significant progress in both general and bilingual applications. Foundational models like LLaMA (Touvron et al. 2023) offer efficient fine-tuning capabilities on vast unlabeled data, while specialized models such as WizardMath (Luo et al. 2023) and MAmmoTH (Yue et al. 2023) enhance mathematical reasoning through innovative training methods and comprehensive datasets. Additionally, models like LLeMMA (Azerbayev et al. 2023)

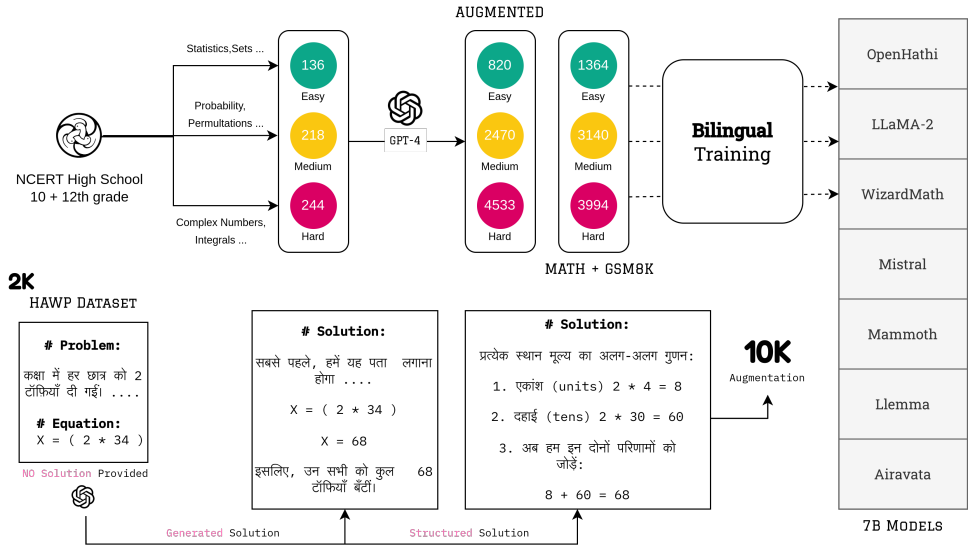

Figure 2: Overall Methodology: The top section illustrates our primary approach, which combines Curriculum Learning and Bilingual Integrated Training. The bottom section depicts the process of applying the decomposition strategy to the HAWP dataset.

and Mistral (Jiang et al. 2023) leverage scientific literature and optimized architectures to achieve advanced mathematical competencies without extensive fine-tuning. In the bilingual domain, models such as OpenHathi-7B Sarvam AI (AI 2023), Hi-NOLIN (Research 2023) from the Pythia, and AIRAVATA (Gala et al. 2024) extend capabilities to English and Hindi by incorporating custom tokenizers, multilingual training datasets, and instruction-tuning techniques. These developments collectively enhance the versatility and applicability of open-source LLMs across diverse linguistic and specialized tasks.

Benchmark Datasets

(Cobbe et al. 2021) introduced the **GSM8K** dataset, which consists of 8,500 grade school math problems that require basic arithmetic operations. These problems are designed to be solvable by proficient middle school students. Similarly, (Hendrycks et al. 2021) released the **MATH** dataset, containing 12,500 complex problems from high school competitions such as AMC 10 and AMC 12, intended for high school students. It covers topics: Algebra, Counting and Probability, Geometry, Intermediate Algebra, Number Theory, Prealgebra, and Precalculus. (Lightman et al. 2023) introduced **PRM800K**, a dataset with 800,000 step-level feedback labels for solutions to MATH (Hendrycks et al. 2021) problems, providing annotations (”Positive,” ”Negative,” or ”Neutral”) to each solution step. For Hindi-speaking students, (Sharma, Mishra, and Sharma 2022) released **HAWP** (Hindi Arithmetic Word Problems), which is the only publicly available dataset of Hindi mathematical questions. It is for grades 1 to 6, and comprises 2,336 basic math word problems requiring a single operator solution. In the next section, we will discuss more on dataset augmentation to create training dataset and their integration in overall process.

Methodology

Decomposition Strategy on HAWP Dataset

To improve the computational accuracy of large language models (LLMs) in arithmetic operations involving large numbers, we propose a Decomposition Strategy for multiplication and division tasks. For multiplication, this involves breaking down the multiplicand into place value components—such as hundreds, tens, and ones—and multiplying each by the other multiplicand. The products are then aggregated to obtain the final result. For division, the dividend is similarly decomposed into segments, each divided by the divisor, with the quotients summed to produce the final answer. This has been proposed to combat the poor calculation skills of open-source language models. In this paper, we focus on introducing and validating the Decomposition Strategy using the HAWP dataset, which contains basic mathematical word problems requiring single-operation calculations. This allows us to clearly demonstrate the strategy’s effectiveness in a controlled, straightforward context. While exploring its application to more complex datasets is an exciting future direction, we have chosen to concentrate on HAWP for now to ensure a thorough and focused evaluation of this novel approach.

We utilized 2,336 Hindi arithmetic problems from the HAWP dataset, covering basic operations like addition, subtraction, multiplication, and division. Since the original dataset lacked solutions, we enhanced it by generating question-answer pairs using GPT-4, which were then carefully reviewed and corrected by five human experts, resulting in the Enhanced HAWP dataset. To evaluate the Decomposition Strategy’s effectiveness, we applied it to the Enhanced HAWP dataset (see Figure 2). Manually solved examples using this strategy were used in few-shot learn-

Model	Add	Sub	Mul	Div
Zero-shot Prompting				
OpenHathi-7B	0.35	0.53	0.44	0.33
LLaMA-2-7B	0.39	0.55	0.33	0.5
LLeMMA-7B	0.49	0.63	0.22	0.17
Mistral-7B	0.49	0.55	0.22	0.25
WizardMath-7B	0.63	0.67	0.22	0.67
Gemini-pro	0.78	0.80	1.0	0.75
GPT-4	0.98	0.93	0.88	0.91
Few-shot Prompting				
OpenHathi-7B	**0.49**	**0.63**	0.34	**0.58**
LLaMA-2-7B	**0.53**	**0.72**	**0.44**	0.5
LLeMMA-7B	**0.82**	**0.9**	**1.0**	**0.67**
Mistal-7B	**0.78**	**0.77**	**0.56**	**0.83**
WizardMath-7B	**0.72**	**0.73**	**0.56**	0.67
Instruction-Tuning (Enhanced HAWP)				
OpenHathi-7B	**0.78**	**0.85**	0.22	0.50
LLeMMA-7B	0.78	0.83	0.67	0.67
WizardMath-7B	**0.96**	**1.0**	**0.78**	**0.75**
Instruction-Tuning (HAWP+Decomposition Strategy)				
OpenHathi-7B	0.78	0.85	0.22	**0.67**
LLeMMA-7B	**0.80**	**0.925**	0.67	**0.83**
WizardMath-7B	0.95	1.0	0.78	**0.83**
Instruction-Tuning (HMQA)				
OpenHathi-7B	**0.82**	0.85	**0.44**	**0.75**
LLeMMA-7B	**0.86**	**0.97**	0.67	0.75

Table 1: Performance of LLMs in Hindi Math questions using decomposition strategy. Bold values indicate improvements from the previous step. Underlined values show the highest performance of SLLMs for each operation.

ing with GPT-4 to modify the remaining solutions in Enhanced HAWP. These refined solutions, with 70% and 30% training/testing split, were then used to fine-tune the models OpenHathi 7B, WizardMath-v1.1 7B, and LLeMMa 7B. The resulting accuracy improvements are shown in Table 1. In our final phase, we focused on exploring the benefits of fine-tuning using an augmented version of the dataset that we previously prepared Decomposition Strategy-enhanced dataset. We expanded the original 2,336 problems to 10,000 using a one-shot prompting technique with GPT-4. These newly generated samples were carefully reviewed by five human experts for accuracy, resulting in the HMQA (Hindi Math Questions-Answers) dataset. We then used this augmented dataset to fine-tune the models OpenHathi 7B, WizardMath-v1.1 7B, and LLeMMa 7B, with the resulting accuracy compared to previous settings in Table 1.

IndiMathQA

We have meticulously curated our own comprehensive math problem dataset, referred to as IndiMathQA, sourcing problems from the official NCERT textbooks[3] used in Indian schools. This dataset contains 598 manually curated math problems and their corresponding solutions. These problems are suited for students in grades 10, 11, and 12, and it encompasses a wide range of problems that vary in complexity and span 14 major mathematical domains, including sets, trigonometry, and the binomial theorem, among others. Appendix provides more details on topic distribution.

LLM Enhancement in Bilingual Mathematics

In this section we demonstrate the strategies used in improving mathematical reasoning skills in Bilingual settings. Our proposed strategies are namely Structured Solution Creation, Curriculum Learning, and Bilingual Training in Hindi and English. We explain the Bilingual Training Dataset Creation in two phases: (i) Classification based on Complexity (required for Curriculum Learning), (ii) Structured Solution Creation and Bilingual Translations. Finally, we demonstrate how we conducted curriculum learning based bilingual fine-tuning on our training datasets.

Classification based on Complexity We have carefully curated a collection of mathematical problems categorized into easy, medium, and hard levels. This collection includes problems from our own dataset as well as from benchmark datasets, such as GSM8K and MATH. Below, we outline the methods we used to classify each problem by its complexity. We utilize additional datasets (GSM8K and MATH) for the sole reason of having more diversity of mathematical topics in our dataset.

IndiMathQA: The IndiMathQA dataset was carefully annotated by a team of five human experts, resulting in 136 easy, 218 medium, and 244 hard questions. To ensure the reliability of these annotations, we calculated the Average Fleiss' Kappa score, which came out to 0.58, indicating low bias and substantial agreement among the annotators. Further details on the annotation process can be found in the Appendix. This dataset was then augmented to a total of 7823 questions with similar concepts (820 easy, 2,470 medium, and 4,533 hard) using the GPT-4 API, which were then reviewed by a team of 5 human experts to correct any errors to ensure accuracy and consistency (See appendix for augmentation prompt details).

GSM8K: GSM8K is a grade-school level mathematics dataset, where all questions are generally low in complexity. However, to ensure precision in our classifications, we used LLama 3 (405B) with prompt engineering to assess and rank the questions according to their complexity, based on various criteria, including Language Understanding, Mathematical Complexity, Reasoning Complexity, Number of Variables, and Conceptual Complexity (details in the appendix). For our experiments, we selected the 700 questions with the lowest complexity as the Easy level questions.

MATH: The MATH dataset features competition-level questions for students in grades 8 through 12, with each question annotated by complexity, ranging from Level 1

[3]NCERT is a major textbook for school students in India.

(easiest) to Level 5 (hardest). For our experiments, we categorized the Level 1 questions in the set as Easy, Levels 2 and 3 as Medium, and the remaining levels as Hard. This classification resulted in 664 Easy, 3,140 Medium, and 3,994 Hard questions.

Structured Solution Generation and Language Translations LLMs often encounter challenges with hallucinations when solving reasoning tasks. In our manual inspection of base model solutions, we noticed that LLMs sometimes became so focused on solving the problem that they overlooked the underlying theoretical principles required for an accurate solution. This observation aligns with findings from (Zheng et al. 2023), which introduces a novel prompting technique that encourages LLMs to step back and ask questions to better understand the background of a problem. Inspired by this, we hypothesized that by fine-tuning our LLMs on solutions that first pause to consider the theoretical framework, we could guide them to produce more accurate responses. Building on this idea, we didn't stop at just providing the theoretical framework; we went a step further. We designed a comprehensive, step-by-step structured solution format for fine-tuning, which we believe will train the LLM to approach reasoning tasks more methodically and with greater accuracy. To achieve this, we transformed the existing solutions in our datasets into a clear, organized format under the following headings: (i) Data Identification, (ii) Problem Analysis, (iii) Theoretical Framework, (iv) Methodology Development, (v) Computation, and (vi) Answer. (see Figure 1)

To guide this process, our team created few-shot examples that illustrate how sample answers should be divided into this structured format. These examples, along with a detailed prompt, were provided to GPT-4, which then generated structured solutions for all problems in our training and testing datasets. A team of 5 human experts then identified and corrected any mistakes in the structured solutions.

Originally, our datasets were in English. After structuring the solutions, we used LLAMA 3 (405B) to translate both the questions and their structured solutions into Hindi. Finally, the English and Hindi versions of GSM8K, MATH, and IndiMathQA are combined on the basis of Easy, Medium, and Hard. This results in a total of 2184 Easy, 5470 Medium, and 8527 Hard problems in our dataset.

Curriculum Learning based Fine-Tuning We apply the technique of Curriculum Learning (Wang, Chen, and Zhu 2021) to SLLMs, hypothesizing that by incrementally increasing the complexity of problems during fine-tuning, we can simulate the natural process of human learning—where mastering simpler tasks paves the way for tackling more challenging ones. Our approach utilizes the Easy and Medium datasets, carefully constructed to cover a diverse range of mathematical topics. Each dataset was divided into 70% for training and 30% for testing, ensuring this split was consistently applied across all problem categories: easy, medium, and hard.

To implement Curriculum Learning, we first train our SLLMs on the Easy dataset, producing a model checkpoint we refer to as SFT_Easy. This checkpoint is then further fine-tuned using the Medium dataset, resulting in the final checkpoint, SFT_Easy+Medium. We evaluate the performance difference between these checkpoints using testing sets from both benchmark datasets and our curated dataset.

We propose a hypothesis that fine-tuning LLMs on a dataset combining identical question-answer pairs in both English and Hindi could enhance the model's ability to understand and reason through math problems in Hindi—a language where the LLM might not be as proficient. Our reasoning is grounded in the idea that by exposing the LLM to parallel data in English, a language it excels in, the model can leverage its strengths in English to build stronger associations and improve its performance in Hindi. To test this hypothesis, our Curriculum Learning-based fine-tuning is conducted in two distinct ways:

1. Training the SLLMs separately on English and Hindi datasets, with results presented in Table 2
2. Employing Bilingual Combined Training, where the model is trained on a combined dataset of both English and Hindi question-answer pairs. The outcomes of this bilingual training are detailed in Table 3 and 4.

For our evaluation of SLLMs in Hindi Math reasoning, we only evaluate performance on the Hindi version of IndiMathQA and the HAWP dataset. For the purpose of clarity, we refer the Hindi version of IndiMathQA as HMKB and the English version as EMKB. (Table 2)

Ablation Study

To comprehensively understand the results and significance of each novel methodology employed in our study, we evaluated performance at every stage. In our experiments with the Hindi dataset, Table 1 shows accuracy metrics attained by base models employing zero-shot and few-shot prompting. The findings underscore a substantial performance enhancement with few-shot prompting compared to zero-shot, demonstrating a notable increase of 20-50% across all operations. This highlights the effectiveness of providing task examples to LLMs. Further, fine-tuning on an enhanced HAWP dataset led to substantial improvements (20-30% in general) in OpenHathi's performance in addition and subtraction tasks, and in WizardMath's performance across all operations. However, LLeMMA-7B's performance declined after fine-tuning. After manual assessment of its responses, we found that it is exhibiting hallucinations in its solutions. This aligns with recent findings that fine-tuning on new knowledge can increase hallucinations (Gekhman et al. 2024). LLeMMA, primarily pre-trained on English mathematical data, showed hallucinations when provided with new Hindi mathematical knowledge. Our novel Decomposition Strategy significantly enhanced LLeMMA's performance, demonstrating that breaking down complex calculations can reduce hallucinations and enhance reasoning skills. Additionally, addressing hallucinations through augmentation of samples proved effective, as shown by the improvements from instruction-tuning on HMQA for both OpenHathi and LLeMMA. A detailed analysis of the benefits of curriculum learning on both Hindi and English datasets is also provided in the following Results & Analysis section.

Models	Settings	English Benchmarks						Hindi Benchmarks			
		GSM8K	MATH	PRM800K	EMKB			Enhanced	HMKB		
					Easy	Medium	Hard	HAWP	Easy	Medium	Hard
LLaMA-7B	Base	33%	22%	27%	36%	28%	21%	19%	17%	11%	8%
LLeMMA-7B	Base	14%	10%	12%	14%	12%	9%	12%	11%	8%	5%
Mistral-7B	Base	37%	23%	29%	39%	30%	24%	25%	22%	14%	10%
MAmmoTH-7B	Base	24%	14%	19%	27%	13%	11%	30%	27%	22%	18%
WizardMath-7B	Base	71%	36%	40%	64%	48%	44%	68%	61%	46%	36%
LLaMA-7B	[SFT_easy]	39%	25%	28%	40%	29%	21%	24%	22%	12%	8%
LLeMMA-7B	[SFT_easy]	21%	11%	12%	21%	13%	9%	15%	14%	9%	5%
Mistral-7B	[SFT_easy]	43%	25%	31%	45%	32%	24%	30%	27%	15%	10%
MAmmoTH-7B	[SFT_easy]	30%	16%	22%	33%	14%	13%	41%	37%	23%	18%
WizardMath-7B	[SFT_easy]	79%	37%	42%	70%	52%	44%	73%	66%	47%	37%
LLaMA-7B	[SFT_easy+medium]	42%	35%	34%	41%	36%	24%	25%	25%	20%	16%
LLeMMA-7B	[SFT_easy+medium]	21%	18%	19%	21%	18%	12%	15%	15%	11%	10%
Mistral-7B	[SFT_easy+medium]	45%	37%	34%	46%	39%	26%	31%	29%	28%	22%
MAmmoTH-7B	[SFT_easy+medium]	33%	25%	30%	34%	21%	15%	42%	40%	32%	26%
WizardMath-7B	[SFT_easy+medium]	**80%**	**45%**	**44%**	**73%**	**64%**	**46%**	**77%**	**69%**	**52%**	**42%**
Bilingual Model Evaluation											
OpenHathi-7B	Base	33%	19%	24%	36%	26%	20%	50%	32%	28%	24%
Airavata-7B	Base	22%	11%	15%	21%	12%	9%	12%	14%	10%	6%
Hi-NOLIN-9B	Base	31%	16%	22%	30%	21%	16%	45%	30%	26%	24%
Closed Source Models											
Gemini 1.0 Pro	Base	75%	39%	38%	68%	60%	43%	81%	72%	60%	48%
GPT-4	Base	91%	57%	70%	92%	90%	81%	93%	91%	83%	70%

Table 2: Performance Comparison of Open-Source and Closed-Source Models on English and Hindi Mathematical Benchmarks

Result & Analysis

In this section, we first examine the impact of Curriculum Learning based fine-tuning in Hindi and English separately. The analysis then explores the results from bilingual combined training. Lastly, we compare the problem-solving capabilities of lightweight open-source models (SLLMs) against closed-source models (LLMs) across different languages and difficulty levels.

Curriculum Learning - English Training

We explore the impact of Curriculum Learning on English Dataset on the performance of SLLMs. In the base setting, the models were fine-tuned on the entire English dataset without distinguishing problem complexity. In this setting, WizardMath-7B demonstrated the highest performance, while LLeMMA-7B exhibited the lowest performance across all benchmarks and our English dataset, EMKB, as shown in Table 2. Following this, the models underwent fine-tuning on a subset of easy problems (SFT_easy), leading to a 4-6% improvement on easy problems and a 6-8% increase on the GSM8K benchmark, indicating effective learning of simpler questions during this phase. However, the improvements on more challenging benchmarks like MATH and PRM800K were modest, with only a 1-2% increase. In the next stage, models were fine-tuned on both easy and medium problems (SFT_easy+medium). This approach yielded a consistent 6% performance increase on medium problems and a 3% improvement on hard problems. These findings (Table 2), suggest that systematically increasing the difficulty of problems enables models to surpass their base setting performance.

Curriculum Learning - Hindi Training

When applying Curriculum Learning to the Hindi datasets, initially, WizardMath-7B led, while LLeMMA-7B lagged on the Enhanced HAWP Benchmark. Fine-tuning on easy problems (SFT_easy) improved performance by 3-5%, but gains on medium and hard problems were minimal. Introducing Curriculum Learning (SFT_easy+medium) led to an additional 2-4% improvement on the benchmark and 3-5% on more difficult problems (Table 2). This stepwise training approach effectively enhanced the models' ability to tackle increasingly complex tasks, demonstrating the value of a structured learning regimen in Hindi datasets.

Curriculum Learning - Bilingual Combined Training

Finally, we tested the performance of SLLMs on full IndiMathQA dataset, covering both Hindi and English versions (Tables 3 and 4). SLLMs were fine-tuned using Curriculum Learning on a bilingual combined training set. As a general trend, all models that went through a combined bilingual training (Tables 3 and 4) performed better on Hindi Benchmarks in comparison to single language fine-tuning (Table 2). This is a remarkable enhacement achieved from our hypothesis that combined fine-tuning on English and Hindi can help improve model's Hindi Mathematical Reasoning. Initially, WizardMath-7B achieved the highest performance, while Airavata-7B had the lowest results (Base Settings: Table 2). Fine-tuning on easy problems (SFT_Easy: Table 3) in both languages led to a consistent 3-5% improvement on easy questions, enhancing the models' ability to generalize across different linguistic contexts. However, improvements

on medium and hard problems were minimal, highlighting the limitations of focusing solely on easy problems. When fine-tuned on both easy and medium problems in both languages (SFT_Easy+Medium: Table 4), the models showed more significant gains, with medium problems improving by 11-18% and hard problems by around 2%. This demonstrates the effectiveness of Curriculum Learning in enhancing problem-solving abilities and leveraging bilingual training.

Fine-Tuning Open-Source Models (SLLMs)

In our evaluation of the performance of open-source models such as LLaMA-7B, LLeMMA-7B, Mistral-7B, MAmmoTH-7B, and WizardMath-7B when fine-tuned on combined both Hindi and English versions of IndiMathQA (HMKB and EMKB) (Tables 3 and 4), we observed that fine-tuning on both languages combined improves the performance on both the languages significantly compared to the gains when fine-tuning on a single language. As shown in Table 3 and 4, fine-tuning on easy problems from both languages led to a marginal 2-3% performance increase on easy problems in both Hindi and English. This improvement is better than the pre-trained models but less substantial than the improvements seen with single-language fine-tuning, as indicated in Table 2. However, Table 3 and 4 further demonstrate that fine-tuning easy and medium problems from both languages resulted in a significant major improvement of 11-18% accuracy.

SLLMs (Lightweight open-source) vs LLMs (closed-source)

WizardMath-7B is the best-performing SLLM in our research. Although GPT-4 performance exceeds even the enhanced performance of WizardMath (Table 2, 3 and 4), through Curriculum Learning (SFT_easy+medium) and Bilingual Parallel Training, WizardMath-7B outperforms Gemini 1.0 Pro in English datasets by about 5% (Table 2, 3 and 4). This improvement highlights the effectiveness of our methodology in enhancing SLLM's problem-solving abilities in English. However, in Hindi datasets, while WizardMath-7B performance is comparable to Gemini Pro, it still lags by approximately 3% across Medium and Hard difficulties, likely because WizardMath is more proficient in solving math problems in English than in Hindi.

English Models vs Bilingual Models

Finally, in this comparative analysis of bilingual models and other open-source models (Tables 2, 3 and 4), we observe that bilingual models perform consistently better across English and Hindi, unlike most open-source models, except for WizardMath-7B. This consistency is likely due to the language-independent nature of mathematical reasoning. However, bilingual models like OpenHathi-7B, which are not pre-trained on mathematical tasks, show only slight improvement after fine-tuning, suggesting limited learning efficiency. The superior performance of WizardMath-7B highlights the importance of pre-training models on mathematical tasks for robust performance across languages.

Models	**IndiMathQA**					
	EMKB			**HMKB**		
	Easy	**Medium**	**Hard**	**Easy**	**Medium**	**Hard**
LLaMA-7B	43%	31%	22%	25%	16%	13%
Llemma-7B	20%	12%	9%	14%	10%	7%
Mistral-7B	48%	33%	24%	30%	21%	16%
Mammoth-7B	36%	15%	13%	40%	28%	23%
WizardMath-7B	**73%**	**64%**	**44%**	**68%**	**46%**	**38%**
Bilingual Models						
OpenHathi-7B	41%	30%	23%	36%	34%	26%
Airavata	23%	14%	11%	16%	11%	9%
Hi-NOLIN	38%	27%	25%	33%	32%	25%

Table 3: Performance of Bilingual Models on IndiMathQA Using SFT_easy Training

Models	**IndiMathQA**					
	EMKB			**HMKB**		
	Easy	**Medium**	**Hard**	**Easy**	**Medium**	**Hard**
LLaMA-7B	44%	35%	23%	29%	24%	19%
Llemma-7B	21%	14%	8%	15%	10%	9%
Mistral-7B	50%	39%	29%	33%	26%	22%
Mammoth-7B	40%	20%	18%	44%	35%	27%
WizardMath-7B	**75%**	**66%**	**47%**	**72%**	**57%**	**45%**
Bilingual Models						
OpenHathi-7B	43%	33%	24%	40%	37%	31%
Airavata	25%	16%	13%	18%	14%	11%
Hi-NOLIN	40%	30%	21%	39%	35%	28%

Table 4: Performance of Bilingual Models on IndiMathQA Using SFT_easy+medium Training

Conclusion

This research developed a Bilingual Math Problem Solver using curriculum learning, query decomposition, and structured solution generation. The Decomposition Strategy improved reasoning by breaking down complex queries, Structured Solution addressed the problem of Hallucinations, while curriculum learning enhanced performance on medium and hard problems. WizardMath-7B consistently outperformed other SLLMs (Lightweight open-source) models and often surpassed closed-source models like Gemini 1.0 Pro with these strategies. Our findings demonstrate that integrating these methodologies significantly enhances the problem-solving capabilities of LLMs. Bilingual Parallel Training (Training in multiple languages) provided diverse problem-solving perspectives, proving more effective than single-language training. This study shows how these diverse methodologies can be used to address issues with LLMs in math problem-solving, and can effectively enhance their performance in Hindi.

Acknowledgments

Dr. Rajiv Ratn Shah is partly supported by the Infosys Center for AI, the Center of Design and New Media, and the Center of Excellence in Healthcare at Indraprastha Institute of Information Technology, Delhi.

References

AI, S. 2023. OpenHathi Series: An Approach To Build Bilingual LLMs Frugally. https://www.sarvam.ai/blog/announcing-openhathi-series. Accessed: 2024-02-15.

Anand, A.; Gupta, M.; Prasad, K.; Goel, U.; Lal, N.; Verma, A.; and Shah, R. R. 2023a. KG-CTG: citation generation through knowledge graph-guided large language models. In *International Conference on Big Data Analytics*, 37–49. Springer Nature Switzerland Cham.

Anand, A.; Gupta, M.; Prasad, K.; Singla, N.; Sanjeev, S.; Kumar, J.; Shivam, A. R.; and Shah, R. R. 2023b. Mathify: Evaluating Large Language Models on Mathematical Problem Solving Tasks.

Anand, A.; Jaiswal, R.; Dharmadhikari, A.; Marathe, A.; Popat, H.; Mital, H.; Nair, A. R.; Prasad, K.; Kumar, S.; Verma, A.; Shah, R. R.; and Zimmermann, R. 2024a. GeoVQA: A Comprehensive Multimodal Geometry Dataset for Secondary Education. *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, 102–108.

Anand, A.; Jaiswal, R.; Dharmadhikari, A.; Marathe, A.; Popat, H. P.; Mital, H.; Prasad, K.; Shah, R. R.; and Zimmermann, R. 2024b. Improving Multimodal LLMs Ability In Geometry Problem Solving, Reasoning, And Multistep Scoring. *arXiv preprint arXiv:2412.00846*.

Anand, A.; Kapuriya, J.; Singh, A.; Saraf, J.; Lal, N.; Verma, A.; Gupta, R.; and Shah, R. 2024c. MM-PhyQA: Multimodal Physics Question-Answering with Multi-image CoT Prompting. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 53–64. Springer Nature Singapore Singapore.

Anand, A.; Nair, A. R.; Prasad, K.; Narayan, V.; Lal, N.; Mahata, D.; Singla, Y. K.; and Shah, R. R. 2024d. Advances in Citation Text Generation: Leveraging Multi-Source Seq2Seq Models and Large Language Models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, 56–64. New York, NY, USA: Association for Computing Machinery. ISBN 9798400704369.

Azerbayev, Z.; Schoelkopf, H.; Paster, K.; Santos, M. D.; McAleer, S.; Jiang, A. Q.; Deng, J.; Biderman, S.; and Welleck, S. 2023. Llemma: An Open Language Model For Mathematics. arXiv:2310.10631.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dave, N.; Kifer, D.; Giles, C. L.; and Mali, A. 2024. Investigating Symbolic Capabilities of Large Language Models. *arXiv preprint arXiv:2405.13209*.

Gala, J.; Jayakumar, T.; Husain, J. A.; Khan, M. S. U. R.; Kanojia, D.; Puduppully, R.; Khapra, M. M.; Dabre, R.; Murthy, R.; Kunchukuttan, A.; et al. 2024. Airavata: Introducing Hindi Instruction-tuned LLM. *arXiv preprint arXiv:2401.15006*.

Gekhman, Z.; Yona, G.; Aharoni, R.; Eyal, M.; Feder, A.; Reichart, R.; and Herzig, J. 2024. Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations? *arXiv preprint arXiv:2405.05904*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Kao, K.-C.; Wang, R.; and Hsieh, C.-J. 2024. Solving for X and Beyond: Can Large Language Models Solve Complex Math Problems with More-Than-Two Unknowns? *arXiv preprint arXiv:2407.05134*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Luo, H.; Sun, Q.; Xu, C.; Zhao, P.; Lou, J.; Tao, C.; Geng, X.; Lin, Q.; Chen, S.; and Zhang, D. 2023. WizardMath: Empowering Mathematical Reasoning for Large Language Models via Reinforced Evol-Instruct. arXiv:2308.09583.

Research, N. 2023. Introducing NOLIN. https://blog.nolano.ai/Hi-NOLIN/. Accessed: 2024-02-15.

Sharma, H.; Mishra, P.; and Sharma, D. 2022. HAWP: a Dataset for Hindi Arithmetic Word Problem Solving. In Calzolari, N.; Béchet, F.; Blache, P.; Choukri, K.; Cieri, C.; Declerck, T.; Goggi, S.; Isahara, H.; Maegaard, B.; Mariani, J.; Mazo, H.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 3479–3490. Marseille, France: European Language Resources Association.

Soviany, P.; Ionescu, R. T.; Rota, P.; and Sebe, N. 2022. Curriculum Learning: A Survey. *Int. J. Comput. Vision*, 130(6): 1526–1565.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, X.; Chen, Y.; and Zhu, W. 2021. A Survey on Curriculum Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 4555–4576.

Yue, X.; Qu, X.; Zhang, G.; Fu, Y.; Huang, W.; Sun, H.; Su, Y.; and Chen, W. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Zhang, H.; Da, J.; Lee, D.; Robinson, V.; Wu, C.; Song, W.; Zhao, T.; Raja, P.; Slack, D.; Lyu, Q.; et al. 2024. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Zheng, H. S.; Mishra, S.; Chen, X.; Cheng, H.-T.; Chi, E. H.; Le, Q. V.; and Zhou, D. 2023. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*.