Breaking Language Barriers in Multilingual Mathematical Reasoning: Insights and Observations

Anonymous ACL submission

Abstract

Existing research predominantly focuses on developing powerful large language models (LLMs) for mathematical reasoning within monolingual languages, with few explorations in preserving efficacy in a multilingual context. To bridge this gap, this paper pioneers exploring and training powerful Multilingual Math Reasoning (xMR) LLMs. Firstly, by utilizing translation, we construct the first multilingual math reasoning instruction dataset, `MGSM8KInstruct`, encompassing ten distinct languages, thus addressing the issue of training data scarcity in xMR tasks. Based on the collected dataset, we propose different training strategies to build powerful xMR LLMs, named `MathOctopus`, notably outperform conventional open-source LLMs and exhibit superiority over ChatGPT in few-shot scenarios. Notably, `MathOctopus`-13B reaches 47.6% accuracy which exceeds ChatGPT 46.3% on MGSM testset. Beyond remarkable results, we unearth several pivotal observations and insights from extensive experiments: (1) When extending the rejection sampling strategy to the multilingual context, it proves effective for model performances, albeit limited. (2) Employing parallel corpora for math Supervised Fine-Tuning (SFT) across multiple languages not only significantly enhances model performance multilingually but also elevates their monolingual performance. This indicates that crafting multilingual corpora can be regarded as a vital strategy for enhancing model performance in a specific language, especially in mathematical reasoning tasks. For instance, `MathOctopus`-7B improves its counterparts that trained on English from 42.2% to 50.8% on GSM8K testset.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Hu et al., 2021; Zeng et al., 2022; OpenAI, 2023; Scao et al., 2022) such as Bloom (Scao et al., 2022) and GPT4 (OpenAI, 2023) have exhibited remarkable performances across a wide array of downstream tasks. Notwithstanding these significant advancements, even the most extensive LLMs are confronted with challenges when faced with mathematical reasoning tasks that necessitate multiple reasoning steps (Gao et al., 2023).

Many recent works focus on using different prompting methods like chain-of-thought (CoT) to solve mathematical problems based on close-sourced LLMs such as ChatGPT and GPT-4. Significantly, LLaMA-Family models (Touvron et al., 2023a,c) have ignited an open-source movement and rapidly reduced the disparity with these closed-source LLMs. Following the line, Yuan et al. (2023a) apply rejection sampling fine-tuning (RFT) for math reasoning LLMs. WizardMath (Luo et al., 2023) advance mathematical reasoning in LLMs through Reinforcement Learning from Evol-Instruct (RLEIF). However, current efforts are primarily focusing on improving the performance of LLMs in English. Although, Shi et al. (2022) propose MGSM testset to explore multilingual math reasoning through in-context learning, training a powerful multilingual mathematical reasoning LLM remains under-explored.

To this end, this paper empirically investigates and enhances the multilingual mathematical reasoning abilities of current open-source LLMs such as LLaMA through supervised learning. Given mathematical problems from different languages, we aspire one single model to correctly answer questions in the respective target language, not just limited to English. However, when scaling out mathematical reasoning to multiple languages, i.e., the task of multilingual mathematical reasoning or **xMR** for short, one main challenge is the training data scarcity in low-resource languages, where no training examples are available. To tackle this challenge, we begin by using ChatGPT to translate the English GSM8K math training dataset into 9 vari-

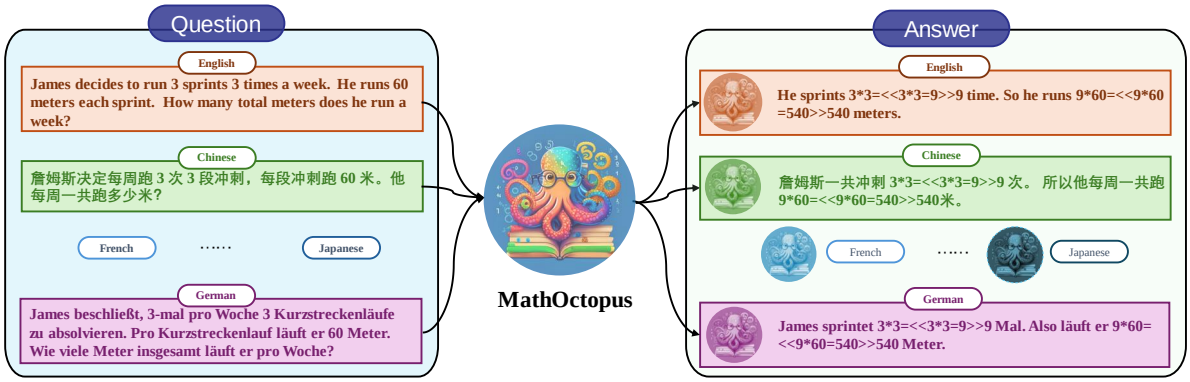


Figure 1: Multilingual Math Instruction Tuning of `MathOctopus`.

ous languages. Concurrently, we employ specific rules to calibrate and align the translated corpora, ensuring the data quality.

The resulting data are used to construct our multilingual math instruction training dataset: `MGSM8KInstruct`, which encompasses instructional data within two distinct settings: *Parallel-training* and *Cross-training*. The Parallel-training setting denotes that both the mathematical queries and the CoT answers derive from the same language. Conversely, the Cross-training setting indicates that the questions are in English, while the corresponding answers are in other languages. The objective of these settings is to develop LLMs are capable of solving mathematical problems coherently, whether presented in English or other target languages, while ensuring adaptability and maintaining rigorous mathematical reasoning across multiple languages. Step further, to conduct a more exhaustive and comprehensive evaluation of the model's multilingual mathematical capabilities, we additionally develop an out-of-domain xMR testset: **MSVAMP**, incorporating 10 languages, based on SVAMP (Patel et al., 2021). This construction involves the employment of machine translation, further refined by meticulous manual calibration.

We then use `MGSM8KInstruct` to supervise fine-tune (SFT) current open-source LLMs including LLaMA-Families ranging from 7B to 33B, as shown in Figure 1. Following training, the models demonstrate exemplary abilities in xMR tasks. We name the resulting models as `MathOctopus`, attributing their adaptability and extensive proficiency in xMR tasks across a variety of languages. For example, `MathOctopus` -7B improves the average performance of LLaMA2-7B on the MGSM dataset across ten languages from 22.6% to 40%. Surprisingly, compared with LLMs trained on the monolingual corpus, `MathOctopus` also shows superior performances when tested in their respective training languages. For instance, `MathOctopus`-7B elevate the accuracy of LLaMA2-7B on the English GSM8K from 42.3% to 50.8%. We posit that this improvement is attributable to the enhanced generalization capability of the model acquired during multilingual training, as well as the reasoning knowledge learned from other languages feeding back into in English.

Subsequently, we delve into an exploration of the influences of SFT data volumes and diverse reasoning paths on the efficacy of the SFT model in multilingual context. Inspired by (Yuan et al., 2023a), we apply multilingual rejection sampling on `MathOctopus` to generate different correct reasoning paths as an augmented training corpus. By incorporating this data into our prior `MGSM8KInstruct` for fine-tuning LLMs, we observe limited enhanced multilingual mathematical reasoning outcomes. We term this training approach Multilingual Rejection Sampling Fine-tuning (xRFT). Experimentally, xRFT can further elevate the LLM's xMR outcomes, but it may potentially compromise the model's generalization ability as the data amount increases.

In summary, our contributions are as follows:

- We construct `MGSM8KInstruct`, a first multilingual math reasoning instruction dataset. Subsequently, MSVAMP, an out-of-domain multilingual mathematical reasoning dataset, is collected, serving as a robust test bed to further assess the robustness of LLMs.

- Based on collected data and different training strategies, we build a series of powerful LLMs `MathOctopus` in xMR tasks. Our model not only significantly improves its reasoning capabilities in low-resource languages compared

to LLaMA but also greatly enhances its performance in English.

- We explore the relationship between model performance and data volume, as well as the impact of different training strategies. One of the most surprising observation is that multilingual SFT could be regarded as a crucial strategy for enhancing mathematical reasoning proficiency in LLMs.

2 Methodology

In this section, we aim to illustrate our method in detail. We first review the problem formulation of multilingual mathematical reasoning. Then we describe the collection process of `MGSM8KInstruct` and MSVAMP datasets. Subsequently, we present our proposed `MathOctopus` and multilingual rejection sampling methods, sequentially.

2.1 Problem Formulation

Commonly, the mathematical reasoning problem-solving task can be defined as $\mathcal{D} = \{Q_i, O_i, A_i\}$, where Q_i is the target math question, $O = \{O_1, O_2, ..., O_k\}$ are answer options if Q_i is a K-way multiple choice problem, A_i is the corresponding ground-truth answer. Given Q_i as inputs, LLMs can directly output answers or a sequence of tokens as intermediate reasoning steps R_i via CoT. Then we can obtain the answer in R_i through regular expression matching.

In this work, we extend mathematical reasoning tasks from monolingual to multilingual contexts: $\mathcal{D}_{en}$ to $\mathcal{D}_{en}...\mathcal{D}_{zh}$. We aspire to enable only one model to successfully solve mathematical problems presented in various languages. That is, given $Q_{i(zh,es,...)}$ in a target language like Chinese, Spanish and etc, the model can furnish correct CoT responses $R_{i(zh,es,...)}$ in a specific language, even when the problems are solely presented in English.

2.2 MGSM8KInstruct

Source Data Prior to going further, the main concern in xMR is data scarcity of the multilingual training corpus. We employ GSM8K (Cobbe et al., 2021), an English dataset comprised of middle grade-school mathematical problems annotated by humans, as our fundamental data source. According to the officially provided solutions, each problem in GSM8K necessitates a resolution process involving between two to eight steps.

Target Languages As for target languages in translation, we choose a set of ten languages that are typologically varied from English (En), covering different language families. Similar with (Shi et al., 2022), the ensemble of languages incorporated in this study comprises Bengali (Bn), Chinese (Zh), French (Fr), German (De), Japanese (Ja), Russian (Ru), Spanish (Es), Swahili (Sw), and Thai (Th). This diverse conglomerate facilitates an exhaustive exploration into the model's adaptability and proficiency amidst divergent linguistic architectures and typologies.

Translation Process Given the reliability and applicability of ChatGPT for translation tasks, we utilize ChatGPT to translate 7473 problems and CoT responses from the English GSM8K training set and their corresponding CoT answers into the target languages. To ensure the quality and consistency of the translations, we adopt the following strategies within the translation prompts:

1. Maintain consistent translations for names of people and places within the sentences.
2. Preserve the mathematical formulas during translation.
3. All numbers must be represented using Arabic numerals to facilitate cross-lingual prediction.
4. To pursue more accurate translations, we provide two translation examples in the prompts for each language.

Verify Strategy Upon inspection of our randomly sampled translations, we find that ChatGPT generally maintains semantic accuracy in translations; however, discrepancies in formula translations may arise. Thus, to uphold consistency and accuracy across multiple languages, we additionally extract all formulas present in the translated answers. If all formulas are calculated correctly and are consistent with those in English, we deem the translation to be accurate and error-free. Note that if errors persist across 5 consecutive translations, we discard that particular case.

This approach ensures a coherent and accurate translation process, allowing for comprehensive evaluation and application in xMR tasks while maintaining linguistic and mathematical integrity. Upon acquiring the translated data, by pairing it with the alpaca-format prompts, we are able to formulate our final training dataset `MGSM8KInstruct`,

Dataset	En	Sw	Zh	Bn	De	Es	Fr	Ja	Ru	Th	Overall
MGSM8KInstruct	7473	7472	7466	6539	7466	7470	7469	7471	7361	7473	**73.6k**
MSVAMP	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	**10k**

Table 1: Data statistics of our MGSM8KInstruct and MSVAMP.

Times	En	Sw	Zh	Bn	De	Es	Fr	Ja	Ru	Th
10	1.5	2.1	1.6	1.2	1.6	1.4	1.2	0.8	1.3	1.4
30	2.5	3.5	2.7	1.9	2.6	2.3	2.0	1.3	2.2	2.3
50	3.8	5.2	4.0	2.9	3.9	3.5	3.0	1.9	3.3	3.4

Table 2: Distinct reasoning paths of each language with different sampling times.

with about 73.6k samples, statistics shown in Table 1. Translation and training prompts are displayed in the Appendix, Table 7 and 9.

2.3 MathOctopus

Training Strategies We then use multilingual query-response pairs in MGSM8KInstruct to supervise fine-tune LLMs, resulting in MathOctopus. As illustrated before, let us delve into a detailed exposition of our diverse training strategies:

- *Parallel-training*, involves filling in the input prompts with questions and answers in the same native language during training. This strategy is like teaching the model to communicate clearly in one language at a time. It helps the model get better at answering questions accurately within the same language, making it more reliable and effective.

- *Cross-training*, refers to our approach during training where we insert English questions and answers in one native language into the input prompts. This approach is like mixing languages in teaching, using English questions and native language answers. It helps the model understand and connect different languages better, making it more versatile and capable of handling multilingual scenarios.

Multilingual Rejection Sampling Prior work (Yuan et al., 2023a) has demonstrated that LLM's performance can be further enhanced by augmenting data through rejection sampling (RFT). Consequently, in this paper, we explore whether the gains imparted by RFT persist in multilingual scenarios. After obtaining the preliminary SFT model, we perform multiple inferences with the SFT model in the MGSM8KInstruct dataset, sampling more diverse and accurate reasoning paths from different languages to integrate into the original dataset. More specifically, we first eliminate samples with incorrect final answers. Subsequently, we extract all the formulas in each reasoning path and validate their accuracy; if all are correct, we consider that reasoning path as correct. We then follow the strategies from (Yuan et al., 2023a) to acquire different correct reasoning paths: a reasoning path is only collected as augmented data if no previously collected path contains identical formulas.

However, the reasoning paths sampled from one single SFT model can be logically non-diverse. Consequently, we anticipate further enhancing the mathematical reasoning performance by leveraging reasoning paths aggregated from different models via multilingual rejection sampling. Considering the cost of prolonged inference, we currently perform 25 inferences for each language from the basic MathOctopus 7B and 13B models respectively, meaning we sample answers for each question in every language 50 times. In our experiments, we fuse all the different reasoning paths generated by the two models to obtain our final xRFT augmented data. We set the temperature as 0.9 and with different seeds to expect the model to generate diverse solutions. The table 2 displays the number of different reasoning paths per question produced in each language over 50 samples.

3 Experiments

In this section, we first review our in-domain evaluation dataset: MGSM, and the collection of the out-of-domain testset: MSVAMP. Subsequently, we introduce multiple baselines in this work. At last, we present the main results and findings of our experiments. We illustrate our experimental setup for training and testing in Appendix B.

3.1 Evaluation Datasets

MGSM The source data of MGSM (Shi et al., 2022) is collected from a subset from GSM8K testset, and then native annotators translate the subset in English into other 10 languages. Each language branch consists of 250 test samples.

MSVAMP We choose SVAMP as our English source data. Given that the answers in the SVAMP

Models	En	Sw	Zh	Bn	De	Es	Fr	Ja	Ru	Th	Avg.
Close-Source LLMs											
ChatGPT-Zero shot	52.0	30.0	44.4	4.4	46.8	52.0	45.6	38.8	37.2	8.0	35.9
ChatGPT-En 2shot	67.2	40.0	52.8	7.6	62.0	61.2	59.2	46.8	50.4	15.6	46.3
GPT4-En 2shot	**80.0**	**64.4**	**70.0**	**17.6**	**73.6**	**71.2**	**72.0**	**71.6**	**64.0**	**40.4**	**62.5**
Open-Source LLMs (7B Model)											
LLaMA 2	43.2	5.2	22.4	3.2	37.2	32.4	34.4	15.2	28.0	4.8	22.6
RFT	44.8	2.8	16.8	2.4	33.6	34.0	34.0	6.8	29.2	2.0	20.6
MAmmoTH	49.6	2.4	17.2	3.6	33.2	32.4	32.8	10.8	26.0	4.8	21.3
WizardMath	47.6	3.4	22.4	2.0	30.4	34.8	30.4	24.0	30.8	4.0	23.0
MathOctopus$^{\mathcal{C}}$	52.0	23.6	31.6	18.8	38.0	39.2	36.4	27.2	33.6	21.6	32.2
xRFT-MathOctopus$^{\mathcal{C}}$	51.2	24.0	33.2	18.8	36.0	41.2	37.6	29.6	36.4	25.2	33.3
MathOctopus$^{\mathcal{P}}$	52.4	**39.2**	38.4	28.8	44.8	42.4	43.6	36.0	39.6	34.4	40.0
xRFT-MathOctopus$^{\mathcal{P}}$	**54.8**	38.4	**45.2**	**33.2**	**43.6**	**45.2**	**38.0**	**35.6**	**48.4**	**36.4**	**41.9**
Open-Source LLMs (13B Model)											
LLaMA 2	50.4	7.6	32.8	6.0	42.8	45.2	40.8	25.2	39.2	6.8	29.7
RFT	52.0	3.6	33.6	3.2	38.4	46.8	44.8	26.4	41.6	4.4	29.5
MAmmoth	56.4	1.6	31.2	3.6	45.6	50.0	39.6	19.2	36.8	5.2	28.9
WizardMATH	52.8	5.6	28.0	6.4	40.4	45.6	42.0	22.0	34.4	5.6	28.3
MathOctopus$^{\mathcal{C}}$	**56.4**	27.2	39.2	24.0	47.6	49.6	47.6	40.4	42.0	24.8	39.9
xRFT-MathOctopus$^{\mathcal{C}}$	53.6	28.0	45.2	21.2	48.0	46.4	46.0	35.2	45.6	28.8	39.8
MathOctopus$^{\mathcal{P}}$	53.2	42.8	48.8	35.2	44.4	48.0	48.4	**43.2**	**47.6**	**46.8**	45.8
xRFT-MathOctopus$^{\mathcal{P}}$	51.6	**46.0**	**51.2**	**42.0**	**49.2**	**53.2**	**49.6**	39.6	**47.6**	46.0	**47.6**
Open-Source LLMs (30-34B Model)											
LLaMA 1	50.8	3.6	27.6	3.2	42.4	44.4	42.4	11.6	38.4	1.2	26.6
RFT	**57.6**	2.4	26.0	4.8	45.6	46.4	44.8	9.2	46.4	4.4	28.8
MathOctopus$^{\mathcal{C}}$	55.6	24.4	36.0	19.2	40.4	51.2	44.4	27.2	37.2	21.6	35.7
xRFT-MathOctopus$^{\mathcal{C}}$	53.6	27.6	34.4	19.2	47.2	47.6	44.8	30.8	38.8	22.8	36.7
MathOctopus$^{\mathcal{P}}$	56.4	46.8	52.0	35.2	47.2	**53.2**	**48.0**	39.2	45.6	41.2	46.5
xRFT-MathOctopus$^{\mathcal{P}}$	51.6	**47.2**	**52.4**	**37.6**	**51.2**	52.8	44.4	**41.6**	**50.0**	**47.6**	**47.6**

Table 3: Model Performances on MGSM testset. MathOctopus$^{\mathcal{P}}$ and MathOctopus$^{\mathcal{C}}$ refer to models trained on *parallel-training* and *cross-training*, separately. We highlight the best results in each language of the same backbone.

only contain the numerical results, we focus solely on translating the questions. To ensure high-quality translations, we use Google Translate System to convert 1,000 samples from the SVAMP testset into ten languages, matching the same languages in our training set. To further guarantee translation fidelity, we back-translate the translated text into English, and recruit three professional annotators manually checking for semantic consistency.

3.2 Baselines

Close-Source LLMs In this paper, We consider two OpenAI's LLMs: **GPT-4** and **ChatGPT** (gpt3.5-turbo) for comparison: (1) We test **ChatGPT** with *zero-shot* prompting where none exemplars are given, but we add "Let's think step by step." at the end of the inputs. (2) As Shi et al. (2022) proves including EN-CoT examples could result in better performances in xMR tasks, we additionally test them with *2 shot EN-coT examples*, which are shown in Appendix Table 10.

Open-Source LLMs For fairness, we primarily compare MathOctopus with several LLaMA-based models, including **LLaMA base**, **RFT**, **MAmmoTH** and **WizardMath**. In this work, LLaMA base denotes trained on GSM8K English corpus; RFT utilizes rejection sampling on English instruction tuning; MAmmoTH is trained based on a variety of math instruction datasets; WizardMath is built on Reinforcement Learning from Evol-Instruct (RLEIF) in math reasoning. As we only consider CoT solutions in this work, we don't use the code version of these models.

3.3 Main Results

Table 3 and Table 4 show the in-domain and out-of-domain test results of different open-source LLMs and several close-source LLMs. We run the evaluation three times and report *average results*. We

Models	En	Sw	Zh	Bn	De	Es	Fr	Ja	Ru	Th	Avg.
					Close-Source LLMs						
ChatGPT-Zero shot	76.1	63.2	72.4	3.1	66.7	69.5	71.9	63.3	62.3	24.4	57.3
ChatGPT-En 2 shot	**81.2**	68.4	78.4	14.4	73.9	74.6	78.2	74.0	70.9	46.0	66.0
GPT4-En 2shot	80.1	**75.7**	**78.9**	**31.2**	**78.1**	**81.5**	**83.9**	**74.8**	**77.9**	**68.1**	**73.0**
					Open-Source LLMs (7B Model)						
LLaMA 2-LoRA	27.6	4.0	12.0	2.0	10.4	18.4	16.8	7.6	11.2	3.2	11.3
LLaMA 2	38.8	17.2	35.2	11.5	39.0	39.2	39.1	31.6	39.1	18.2	30.9
RFT	42.7	14.9	34.9	7.7	40.8	42.5	41.5	33.9	39.5	16.9	31.5
MAmmoTH	45.1	4.2	26.8	4.3	39.6	42.9	39.9	26.7	33.7	6.3	27.0
WizardMath	48.5	10.3	36.3	16.1	39.2	44.8	37.7	37.9	37.4	17.0	32.5
MathOctopus$^{\mathcal{C}}$	49.2	36.6	**43.6**	30.2	**48.6**	46.8	46.4	42.5	**46.7**	34.0	42.5
xRFT-MathOctopus$^{\mathcal{C}}$	**49.9**	37.7	43.3	**32.9**	46.5	**47.6**	**47.3**	42.7	46.6	36.2	**43.1**
MathOctopus$^{\mathcal{P}}$-LoRA	30.4	15.2	23.6	10.4	22.8	24.8	26.4	18.0	22.0	14.8	20.8
MathOctopus$^{\mathcal{P}}$	46.5	40.1	42.5	29.1	43.5	45.4	46.0	42.5	45.4	35.7	41.7
xRFT-MathOctopus$^{\mathcal{P}}$	46.8	**42.3**	43.2	32.8	43.1	44.5	45.3	**43.2**	42.1	**40.5**	42.4
					Open-Source LLMs (13B Model)						
LLaMA 2	50.9	19.8	43.3	13.9	46.2	46.1	47.8	41.8	47.8	23.4	38.1
RFT	47.1	19.4	42.3	12.2	45.1	45.6	45.2	42.4	46.5	24.8	37.1
MAmmoth	53.4	12.9	47.7	5.0	52.3	53.9	53.8	42.2	50.7	13.7	38.6
WizardMATH	56.3	12.5	37.0	13.7	48.7	50.4	49.4	29.5	43.8	16.3	35.8
MathOctopus$^{\mathcal{C}}$	**56.6**	40.4	49.0	30.3	**50.9**	**54.2**	**54.7**	**46.3**	**52.4**	35.7	**47.1**
xRFT-MathOctopus$^{\mathcal{C}}$	52.9	41.9	**49.2**	34.1	50.5	52.8	51.5	45.8	50.2	35.7	46.5
MathOctopus$^{\mathcal{P}}$	50.7	43.4	42.6	31.8	48.4	49.4	50.6	41.1	46.9	39.3	44.4
xRFT-MathOctopus$^{\mathcal{P}}$	44.6	**43.4**	46.4	**34.2**	47.7	48.2	49.9	43.1	48.2	**39.5**	44.5
					Open-Source LLMs (30-34B Model)						
LLaMA 1	49.0	9.3	37.5	3.7	44.1	45.6	44.3	27.0	43.1	8.4	31.2
RFT	46.8	11.5	36.6	6.0	46.1	46.8	46.7	31.1	44.9	9.9	32.6
MathOctopus$^{\mathcal{C}}$	51.5	42.1	46.2	23.2	**50.5**	52.1	**52.9**	42.2	**50.5**	33.4	44.5
xRFT-MathOctopus$^{\mathcal{C}}$	48.1	42.8	43.6	23.3	48.7	50.0	48.9	43.4	44.6	35.5	42.9
MathOctopus$^{\mathcal{P}}$	**56.4**	**46.8**	**52.0**	35.2	47.2	**53.2**	48.0	39.2	45.6	41.2	**46.5**
xRFT-MathOctopus$^{\mathcal{P}}$	48.0	42.3	46.1	**36.2**	47.5	48.5	48.3	**45.8**	47.2	**41.2**	45.1

Table 4: Model Performances on MSVAMP testset. MathOctopus$^{\mathcal{P}}$ and MathOctopus$^{\mathcal{C}}$ refer to models trained on *parallel-training* and *cross-training*, separately.

can draw the key observations and insights from the tables:

Cross-Test		
Models	MGSM8K	MSVAMP
MathOctopus$^{\mathcal{P}}$-7B	44.4	47.8
MathOctopus$^{\mathcal{C}}$-7B	**47.0**	**54.2**
MathOctopus$^{\mathcal{P}}$-13B	47.8	45.2
MathOctopus$^{\mathcal{C}}$-13B	**54.2**	**51.5**
MathOctopus$^{\mathcal{P}}$-33B	46.5	46.6
MathOctopus$^{\mathcal{C}}$-33B	**53.3**	**49.4**

Table 5: Average performances of MathOctopus$^{\mathcal{C}}$ and MathOctopus$^{\mathcal{P}}$ under the *Cross-Test* settings.

LLMs struggle in xMR tasks, especially in low-resource languages. From the tables, current open-source LLMs still suffer in xMR in terms of in-domain and out-of-domain testing. For instance, LLMs with 7B-level only achieve about 20%-23% accuracy in MGSM. Another conspicuous observation is the stark performance contrast of existing models when dealing with high-resource versus low-resource languages. This discrepancy can be largely attributed to the diminutive representation of low-resource languages in their foundational corpus. In contrast, our model adeptly rectifies this shortcoming, as evidenced by its enhanced performance in languages like Thai and Swahili.

Performance Superiority of MathOctopus. The proposed MathOctopus, whether in cross-training or parallel-training, both significantly outperforms other open-source LLMs by a large margin. For instance, when operating at the 7B-level, our model can boost the LLaMA model's efficacy from a mere 22.6% to a commendable 41.9% on MGSM. Furthermore, this superiority still remains as the model size escalates. Surpris-

ingly, MathOctopus$^{\mathcal{P}}$-13B even surpasses ChatGPT on MGSM. Another interesting finding is that **MathOctopus$^{\mathcal{P}}$ performs better in the in-domain test while MathOctopus$^{\mathcal{C}}$ shows better generalization ability in the out-of-domain test**, proving their unique advantages, separately.

Models	GSM8K	SVAMP
LLaMA 2-7B	42.4	38.3
MathOctopus$^{\mathcal{P}}$-7B	49.3	46.8
MathOctopus$^{\mathcal{C}}$-7B	**50.8**	**49.3**
LLaMA 2-13B	51.0	50.9
MathOctopus$^{\mathcal{P}}$-13B	55.5	52.1
MathOctopus$^{\mathcal{C}}$-13B	**56.6**	**56.6**
LLaMA 1-33B	50.0	49.0
MathOctopus$^{\mathcal{P}}$-33B	**56.0**	**52.5**
MathOctopus$^{\mathcal{C}}$-33B	53.7	51.5

Table 6: Model performances on English datasets.

Marginal Impact of xRFT. The xRFT's contribution to model enhancement appears to be somewhat circumscribed. Its potency wanes, particularly in out-of-domain test scenarios. In tests within MGSM, the xRFT's contribution to MathOctopus$^{\mathcal{P}}$ hovers around a modest 1%-2% average uplift. However, this figure dips below 1% in MSVAMP testset. More intriguingly, the augmentative effect of xRFT on MathOctopus$^{\mathcal{C}}$ seems even more subdued, with potential performance deterioration observable on the MGSM.

3.3.1 Training-Testing Consistencies

As shown in the tables, equally significant is the discernible performance disparity between models employing MathOctopus$^{\mathcal{C}}$ versus MathOctopus$^{\mathcal{P}}$, when subjected to the same quantum of training data: MathOctopus$^{\mathcal{P}}$ leads to better in-domain test performances compared with MathOctopus$^{\mathcal{C}}$. A deeper delve into this phenomenon revealed an incongruity between the training and testing environments. Specifically, in the MGSM and MSVAMP testsets, the data adheres to a parallel structure, where both the query and its corresponding response are couched in an identical language—a configuration we term *parallel-test*. This is consistent with the parallel-training settings.

However, in our fine-grained experiments, as presented in Table 5 when the testing environment mimics the cross-training format, dubbed *cross-test*, MathOctopus$^{\mathcal{C}}$ demonstrate a superior performance trajectory compared to MathOctopus$^{\mathcal{P}}$ when model sizes scale from 7B to 33B. This underlines the pivotal importance of ensuring a **harmonious alignment between training and testing data formats** for optimizing LLMs' multilingual reasoning capability.

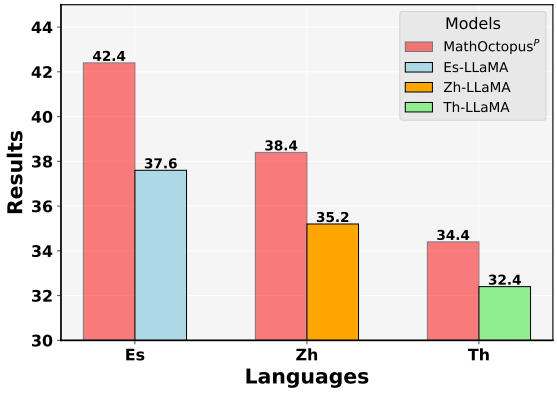
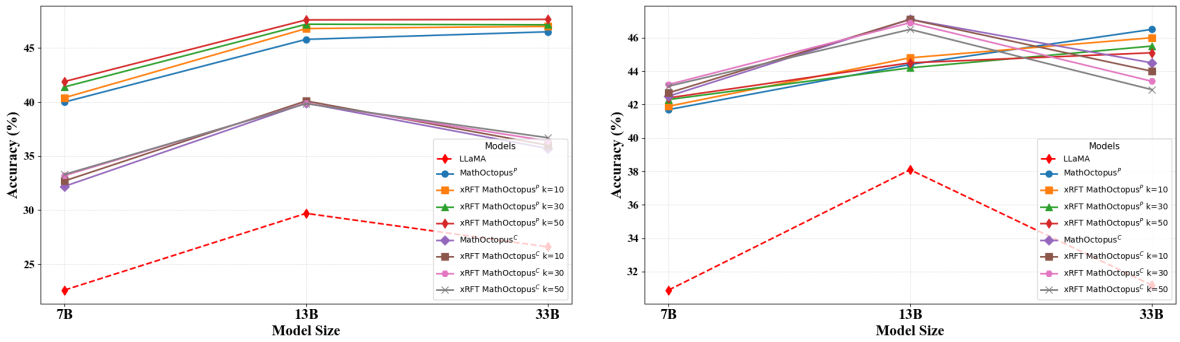


Figure 2: Performances of 7B-models on three languages subset from MGSM.

3.3.2 Multilingual SFT can generally benefit Monolingual SFT

One of the most exciting observations in our experiments is that **MathOctopus could significantly improve the performances in English.** Table 6 presents the corresponding results of LLaMA and MathOctopus on English GSM8K and SVAMP testsets. First and foremost, both MathOctopus$^{\mathcal{P}}$ and MathOctopus$^{\mathcal{C}}$ have substantially enhanced the performance of LLaMA2 across the two datasets. Secondly, MathOctopus$^{\mathcal{C}}$ emerges as the superior performer (e.g., 50.8% vs. 49.3%, 49.3% vs. 46.8% with 7B-level). We surmise that this can be attributed to the cross-training paradigm, where during training, the model predominantly encounters data with English prompts requiring reasoning in other languages. This exposure, intensifies the model's proficiency in English comprehension, providing it with a distinctive edge.

More broadly, does this situation persist in other languages as well? To explore this, we randomly select training sets for three languages from the training dataset: Spanish and Chinese, as well as the low-resource language Thai, and used their respective corpora to train three models, which we named Es-LLaMA, Zh-LLaMA, and Th-LLaMA, respectively. Figure 2 separately illustrates the test results of several models in their respective training languages. We observe that our model still surpasses the results of the monolingual SFT models in their native training languages. This suggests that, at least in the task of math reasoning, **multilingual SFT can be considered a superior training strategy to monolingual SFT, effortlessly elevating the model's performance in its native language.**



(a) Models Performances on MGSM **(b) Models Performances on MSVAMP**

Figure 3: Model performances of MathOctopus with different xRFT data volume.

4 Discussion

In this section, we aim to delve deeper through a series of meticulous experiments designed to address the following inquiries:

- **Q1**: The impact of xRFT: especially the implications of varying xRFT data quantities on the efficacy of the model?
- **Q2**: When solely utilizing corpus of specific languages from MGSM8KInstruct for model training, can we still witness an enhancement in other unseen languages? (See Answer in Appendix C)
- **Q3**: Variations of the training strategy: Step further, instead of strictly restricting the questions to English, we can propose forming pairings from the available corpus of 10 languages in MGSM8KInstruct, thus allowing both the questions and answers to span diverse combinations. How does the model fare under such conditions? (See Answer in Appendix C)

4.1 RQ1: Influence of xRFT Data Volume

In this component, we probe the impact of varying data quantities introduced by different xRFTs on model performance. Figures 3 (a) and (b) respectively illustrate the performances of models across two datasets under xRFT with sampling counts k set to {10, 30, 50}. From the visuals, it's discernible that for MathOctopus$^{\mathcal{P}}$, employing larger training corpus generally augments performance in most cases, a trend particularly pronounced in the MGSM dataset. However, these gains remain modest, especially when the backbone model becomes more performant. In contrast, MathOctopus$^{\mathcal{C}}$ demonstrates only marginal improvements attributable to xRFT, and intriguingly, its efficacy on the MSVAMP dataset seems to wane as the k value escalates.

This suggests that while xRFT introduces various reasoning paths, its contribution to tasks like multilingual math reasoning remains circumscribed. A plausible rationale is that during the multilingual SFT phase, distinct linguistic versions of the same solution might already be construed as diverse reasoning paths. Hence, multilingual SFT can essentially be viewed as a variant of the monolingual SFT's RFT. Building upon the foundation of multilingual SFT, supplementary benefits conferred by xRFT appear to be minuscule and might even usher in model overfitting.

5 Conclusion

In this paper, we pioneer to explore training multilingual mathematical LLMs. To address the data scarcity in low-resource languages, We first collect the first multilingual math reasoning instruction dataset, named MGSM8KInstruct, consisting of ten various languages. The models, trained on the MGSM8KInstruct with different training strategies, named MathOctopus, show superior performances compared to other open-source LLMs. We prove that MathOctopus with *parallel-training* could achieve better in-domain test results while MathOctopus with *cross-training* present better robustness in our collected out-of-domain testset, MSVAMP. We investigate the influences of the multilingual rejection sampling strategy, observing it contributes marginal impact to xMR tasks. Based on our extensive and fine-grained experiments, we also draw several key conclusions, where the most exciting one is that creating aligned bilingual question-answer corpora can be regarded as an important way to improve model performance in its native language, especially in tasks that require mathematical reasoning. Within the picture of our future work, we will explore more methodologies and diverse parallel corpus in training xMR LLMs which may involve the RLHF and DPO.

Limitations

In this work, we still leave several parts to underexplore, which may also contribute to building effective xMR LLMs:

- Developing `MathOctopus` based on larger size LLMs, including LLaMA 2-70B and LLaMA-Coders, which is a future work in our following experiments.
- Currently, we only apply xRFT to 7B and 13B models due to the high cost of inferencing. We also will conduct xRFT to more performant models, further investigating its efficiency.
- We are still not very clear whether including more languages in `MGSM8KInstruct` could benefit current models, which will discussed in our next version.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Nuo Chen, Linjun Shou, Tengtao Song, Ming Gong, Jian Pei, Jianhui Chang, Daxin Jiang, and Jia Li. 2023a. Structural contrastive pretraining for cross-lingual comprehension. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2042–2057, Toronto, Canada. Association for Computational Linguistics.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023b. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. In *ACL (industry)*, pages 37–42. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1896–1907. Association for Computational Linguistics.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Ansong Ni, Jeevana Priya Inala, Chenglong Wang, Alex Polozov, Christopher Meek, Dragomir Radev, and Jianfeng Gao. 2023. Learning math reasoning from self-sampled correct and partially-correct solutions. In *The Eleventh International Conference on Learning Representations*.

OpenAI. 2023. Gpt-4 technical report.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2023. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023c. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Yixuan Weng, Minjun Zhu, Shizhu He, Kang Liu, and Jun Zhao. 2022. Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Hanwei Xu, Yujun Chen, Yulun Du, Nan Shao, Yanggang Wang, Haiyu Li, and Zhilin Yang. 2022. Zeroprompt: Scaling prompt-based pretraining to 1,000 tasks improves zero-shot generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 4235–4252. Association for Computational Linguistics.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023a. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023b. Rrhf: Rank

responses to align language models with human feedback without tears.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023. Solving math word problems via cooperative reasoning induced language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4471–4485, Toronto, Canada. Association for Computational Linguistics.

A Related Works

Math Reasoning with LLMs A pivotal metric for assessing the efficacy of LLMs is their capability in addressing intricate reasoning challenges, exemplified by mathematical reasoning tasks (Scao et al., 2022; Cobbe et al., 2021; Zhou et al., 2022; Weng et al., 2022; Chen et al., 2023b). Rather than yielding direct, definitive answers, prior research has illustrated that by employing a variety of prompting techniques, such as Chain-of-Thought (CoT) prompting (Wei et al., 2022), LLMs can be guided through step-by-step reasoning, resulting in significant improvements in performance across an array of diverse reasoning tasks. Imani et al. (2023) propose the generation of multiple algebraic expressions or Python functions to solve the same mathematical problem, aiming to explore a broader spectrum of potential solutions. Additionally, Li et al. (2023) introduce a step-aware verifier to scrutinize the reasoning steps in COT, thereby enhancing the model's reasoning capabilities. Another effective approach, Self-Consistency (Wang et al., 2022), combines a wider range of solutions and derives a final answer by aggregating them to obtain the most consistent response. Meanwhile, several scholarly works have incorporated the concept of rejection sampling, in conjunction with various other techniques, to curate a more diverse set of sampled reasoning paths for the purpose of fine-tuning data augmentation (Huang et al., 2022; Zelikman et al., 2022; Ni et al., 2023; Zhu et al., 2023; Bai et al., 2022; Yuan et al., 2023b; Dong et al., 2023; Touvron et al., 2023b; Song et al., 2023). Following the line, Yuan et al. (2023a) utilize rejection sampling to augment the data volume for fine-tuning math reasoning LLMs.

Instruction Tuning with LLMs Instruction tuning serves as a pivotal component within the developmental frameworks of language models, with its primary function being to orient LLMs towards objectives that are more congruent with human preferences and functional applications. The academic discourse on instruction tuning is notably concentrated on amplifying the versatile instructional capabilities of LLMs. This discourse is particularly exemplified by pioneering studies such as UnifiedQA (Khashabi et al., 2020), Zero-Prompt (Xu et al., 2022), FLAN (Chung et al., 2022), and T0 (Sanh et al., 2022). These studies have embarked on an exploration into the generalization capabilities of LLMs. Following these, FLAN-v2 (Longpre et al., 2023) further investigated the impact of scaling instructional datasets on model performance. Recent innovations in this domain are veering towards employing synthetic instruction following data, distilled from models like GPT-3/4 (OpenAI, 2023), to align open-source LLMs. Recently, several works have utilized instruction tuning for training math LLMs. Yuan et al. (2023a) propose RFT in math reasoning, and WizardMath (Luo et al., 2023) implements the "evol-instruct reinforcement learning" methodology (RLEIF), which is directed towards the refinement of prevailing math instruction data. However, almost all of them aim to improve mathematical reasoning in English, leaving multilingual mathematical reasoning less explored. This paper aims to fill this gap by exploring effective methods for training robust LLMs in multilingual mathematical reasoning.

B Experimental Setup

Training and Testing In this work, we use open-source LLaMA-2 7B to 13B and LLaMA-1 33B as backbone models, allowing us to build `MathOctopus` in multiple scales. Our codes are based on DeepSpeed and Huggingface Library. For all models, we set the learning rate, epochs and max length as 2e-5, 3 and 512. The batch sizes are set to 8, 4, 2 when models scale from 7B to 33B. During testing, we set the maximum output token as 512 with temperature as 0.0 to keep stable performances. We keep the same prompt in Table 7 for testing `MathOctopus`. Please refer to Section 3.3 for xRFT settings.

C Discussion

C.1 RQ2: Targeted Language Training: Limited Broader Linguistic Reasoning Gains

Our exploration into model training with select languages posits a notable inquiry: Can training with a subset of languages enhance mathematical reasoning across all languages? Engaging two high-resource languages, Spanish and Chinese, and two low-resource languages, Thai and Swahili, for mixed training sessions reveals pivotal insights. Seen in Table 8, while stark performance enhancements are witnessed in trained languages, notably in low-resource ones like Thai (surging from 4% to 36.8%), the model's efficacy varies in languages that are unseen in training: While there are improved outcomes in certain languages like Japanese

Input Prompts	Below is an instruction that describes a task. \n Write a response that appropriately completes the request in { *language* }. Please answer in { *language* }.\n \n ### Instruction: \n {*query*}\n\n ### Response:

Table 7: Training and testing prompts in our experiments.

Models	En	Sw	Zh	Bn	De	Es	Fr	JA	Ru	Th	Avg.
LLaMA 2 (En)	43.2	5.2	22.4	**3.2**	**37.2**	32.4	34.4	15.2	28.0	4.8	22.6
MathOctopus$^{\mathcal{P}}$ (En-Zh-Es)	44.0	3.6	**34.4**	3.2	33.6	**41.2**	**36.8**	**25.2**	**30.4**	4.0	25.6
MathOctopus$^{\mathcal{P}}$ (En-Sw-Th)	**46.0**	**34.4**	27.6	2.4	31.2	35.2	32.4	22.4	27.2	**36.8**	**29.6**

Table 8: Model Performances on MGSM test set. **MathOctopus**$^{\mathcal{P}}$ (En-Zh-Es) refers to we only train **MathOctopus**$^{\mathcal{P}}$ in three languages: English, Chinese and Spanish. Similarly, **MathOctopus**$^{\mathcal{P}}$ (En-Sw-Th) means the **MathOctopus**$^{\mathcal{P}}$ trained in English, Swahili and Thai.

and French, a corresponding decline is witnessed in others, such as German and Russian. This phenomenon might predominantly stem from the disparities in grammatical structures across different languages (Chen et al., 2023a).

C.2 RQ3: Training Variability: Beyond English-centric Questions

Beyond the two training strategies explored in Section 3.3, we further probe alternative approaches to discern their influence on model performance. Hence, we examine two additional strategies: 1) A **Mix-Training** approach, where cross-training and parallel-training data are amalgamated for training; 2) An expansive **Mix-All** method that not only extends cross-training but also randomly pairs two languages from the MGSM8KInstrucT, thereby permitting questions and answers in the training data to traverse various linguistic combinations, effectively amplifying the original training data volume tenfold. The resulting models obtained by the above strategies are called MathOctopus$^{\mathcal{M}}$ and MathOctopus$^{\mathcal{M}}$-All, separately.

From the right-side Figure 4, it is evident that despite these two new training strategies respectively doubling and amplifying training volumes tenfold compared to original parallel-training and cross-training strategies, they do not surpass the results of MathOctopus$^{\mathcal{P}}$. Furthermore, the outcomes from MathOctopus$^{\mathcal{M}}$-All slightly underperform MathOctopus$^{\mathcal{M}}$. Such a phenomenon may arise because, although "mix-all" and "mix-training" expand the original data, the pre-existing data volume already suffices for the model to learn alignment and reasoning capabilities across different languages. An additional, rudimentary data expansion potentially induces overfitting, subsequently

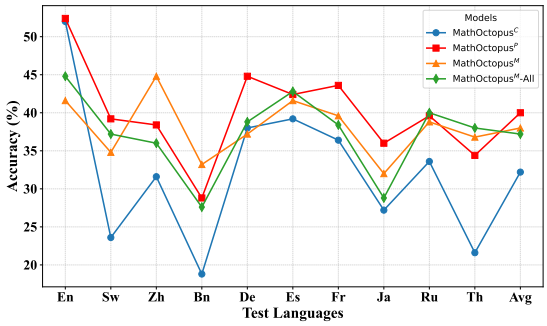


Figure 4: Performances of MathOctopus-7B with different strategy on MGSM.

diminishing model performance.

C.3 Prompts for ChatGPT Translation

Table 9 shows the translation prompt when we utilize ChatGPT to translate the data from English to other languages. When applying this prompt, we just need to replace the placeholder symbol lang with the specific language. Here, we use the Spanish translation examples in the prompt.

C.4 Generated Cases by MathOctopus

Table 11 and Table 12 presens several generated examples in various languages by LLaMA and our models.

Prompts: You are a professional {lang} translator and spelling corrector. Please translate the given math question and its reasoning path into {lang}.
Below are examples:
Q: Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?
P: Babysitting is $12/hour = $12/60 minutes = $«12/60=0.2»0.2 per minute. Weng did babysitting for 50 minutes, so she earned $0.2 x 50 = $«0.2*50=10»10.
T-Q: Weng gana 12 dólares la hora por cuidar niños. Ayer cuidó niños durante 50 minutos. ¿Cuánto ganó?
T-P: Cuidar niños cuesta $12/hora = $12/60 minutos = $«12/60=0.2»0.2 por minute. Weng cuidó niños durante 50 minutos, por lo que ganó $0.2 x 50 = $«0.2*50=10»10.
Q: Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?
P: Since today is the last day of the month, Julie would like to finish an entire book before tomorrow. She has read a total of 12 + 24 = «12+24=36»36 pages in two days. There are 120 - 36 = «120-36=84»84 pages left to be read. Hence, Julie should read 84/2 = «84/2=42»42 pages tomorrow.
T-Q: Julie está leyendo un libro de 120 páginas. Ayer pudo leer 12 páginas y hoy leyó el doble que ayer. Si quiere leer la mitad de las páginas restantes mañana, ¿cuántas páginas debería leer?
T-P: Como hoy es el último día del mes, a Julie le gustaría terminar un libro completo antes de mañana. Ha leído un total de 12 + 24 = «12+24=36»36 páginas en dos días. Quedan 120 - 36 = «120-36=84»84 páginas por leer. Por lo tanto , Julie debería leer 84/2 = «84/2=42»42 páginas mañana.
Please Keep in mind that:
(1) keep the translations consistent for names of people and places within the sentences.
(2) Preserve the mathematical formula within the "« »" brackets when translating.
(3) You must translate the text into {lang}.
(4) You must follow the output format with: "T-Q:... T-P:..."

Table 9: Translation prompts in our experiments.

Prompts: Answer the following math probelm step by step in {lang}.
Below are examples:
Q: Julia played tag with 18 kids on Monday. She played tag with 10 kids on Tuesday. how many more kids did she play with on Monday than on Tuesday?
A: Let's think step by step. Julia playsed tag with 18 kids on Monday and 10 kids Tuesday, separately. So the amount of kids that she played with on Monday than on Tuesday is 18-10=8. So the answer is 8.
Q: Jack had 9 action figures and 10 books on a shelf in his room. later he added 7 more action figures to the shelf. how many more action figures than books were on his shelf ?
A: Let's think step by step. The amount of action figures that Jack had is 9+7=16. And Jack had 10 books. So the amount of action figures than books on his shelf is 16-10=6. So the answer is 7.

Table 10: 2-shot prompts of ChatGPT and GPT-4 in our experiments.

Generated Cases in Chinese	
Query	珍妮特的鸭子每天下16 颗蛋。她每天早上早餐时吃3 颗，每天用4 颗为自己的朋友做松饼。剩下的鸭蛋她每天拿去农贸市场卖，每颗新鲜鸭蛋卖2 美元。她每天在农贸市场赚多少钱?
LLaMA 2	珍妮特每天吃早餐3颗鸭蛋，剩下的鸭蛋用于做松饼4颗，剩下的鸭蛋卖在农贸市场2颗，珍妮特每天在农贸市场赚2*2=«2*2=4»4美元。珍妮特每天卖4 颗鸭蛋，所以她每天在农贸市场赚4*4=«4*4=16»16 美元。✗
MathOctopus	珍妮特每天吃掉3 + 4 = «3+4=7»7 颗鸭蛋。鸭子每天下16 颗蛋，所以剩下16 - 7 = «16-7=9»9 颗蛋。珍妮特每天卖掉9 颗蛋，每颗蛋卖2 美元，所以她每天赚9 * 2 = «9*2=18»**18**。✓
Answer	**18**
Query	图卢兹的绵羊数量是查尔斯顿的两倍。查尔斯顿的绵羊数量是西雅图的4 倍。如果西雅图有20 只绵羊，图卢兹、查尔斯顿和西雅图一共有多少只绵羊?
LLaMA 2	查尔斯顿有2 * 20 = «2*20=40»40 只绵羊。图卢兹有40 * 2 = «40*2=80»80 只绵羊。西雅图和查尔斯顿和图卢兹一共有20 + 40 + 80 = «20+40+80=140»140 只绵羊。✗
MathOctopus	查尔斯顿有4 * 20 = «4*20=80»80 只绵羊。图卢兹有2 * 80 = «2*80=160»160 只绵羊。总共，三个城市有20 + 80 + 160 = «20+80+160=260»**260** 只绵羊。✓
Answer	**260**
Generated Cases in Japanese	
Query	リチャドは15建てのマンションに住んでいる。各には8部屋あり、建物の3/4が入居されている。建物で未入居の部屋は全部で何室あるか?
LLaMA 2	15建てのマンションに8部屋ある各は15*8=«15*8=120»120部屋になる。建物の3/4が入居されているので、入居されている部屋数は120*3/4=«120*3/4=90»90部屋になる。未入居の部屋は90-8=«90-8=82»82部屋である。✗
MathOctopus	15建てのマンションに8部屋あるため、リチャド住む15建てマンションの部屋数は8*15=«8*15=120»120部屋建物の3/4が入居されているため、未入居の部屋は120*1/4=«120*1/4=30»30部屋。✓
Answer	**30**
Query	トゥルズはチャルストンに比べ2倍の数の羊をっています。チャルストンはシアトルに比べ4倍の数の羊をっています。シアトルがっている羊の数が20だったとしたら、トゥルズ、チャルストン、シアトルがっている羊の数は何でしょうか?
LLaMA 2	トゥルズはチャルストンに比べ2倍の数の羊をっているので、トゥルズがっている羊の数は2*4=«2*4=8»8です。チャルストンはシアトルに比べ4倍の数の羊をっているので、チャルストンがっている羊の数は4*20=«4*20=80»80です。シアトルがっている羊の数が20だったとしたら、トゥルズ、チャルストン、シアトルがっている羊の数は80+8+20=«80+8+20=118»118です。✗
MathOctopus	シアトルが20の羊をっているなら、チャルストンは4*20 = «4*20=80»80の羊をっています。トゥルズはチャルストンの2倍の数の羊をっているので、160の羊をっています。合すると、トゥルズ、チャルストン、シアトルは20+80+160 = «20+80+160=260»**260**の羊をっています。✓
Answer	**260**

Table 11: Generated responses compared between LLaMA and ours within 7B-level on MGSM.

Generated Cases in German	
Query	Toula ging in die Bäckerei und kaufte verschiedene Arten von Gebäck. Sie kaufte 3 Dutzend Donuts, die 68 US-Dollar pro Dutzend kosten, 2 Dutzend Mini-Cupcakes, die 80 US-Dollar pro Dutzend kosten und 6 Dutzend Mini-Cheesecakes für 55 US-Dollar pro Dutzend. Wie hoch waren die Gesamtkosten?
LLaMA 2	3 Dutzend Donuts kosteten 3 × 68 = «3*68=204»204 US-Dollar.2 Dutzend Mini-Cupcakes kosteten 2 × 80 = «2*80=160»160 US-Dollar.6 Dutzend Mini-Cheesecakes kosteten 6 × 55 = «6*55=330»330 US-Dollar.Die Gesamtkosten für alle drei Arten von Gebäck waren 204 + 160 + 330 = «204+160+330=794»794 US-Dollar. ✗
`MathOctopus`	Toula kaufte 3 Dutzend Donuts für 68 US-Dollar pro Dutzend, also kosten sie 3 * 68 = 204 US-Dollar.Toula kaufte 2 Dutzend Mini-Cupcakes für 80 US-Dollar pro Dutzend, also kosten sie 2 * 80 = 160 US-Dollar.Toula kaufte 6 Dutzend Mini-Cheesecakes für 55 US-Dollar pro Dutzend, also kosten sie 6 * 55 = 330 US-Dollar.Die Gesamtkosten betragen 204 + 160 + 330 = 694 US-Dollar. ✓
Answer	**694**
Generated Cases in French	
Query	Jacques fait des séries de 3 sprints 3 fois par semaine. Il court 60 mètres à chaque sprint. Au total, combien de mètres court-il par semaine ?
LLaMA 2	Jacques fait 3 * 3 = «3*3=9»9 séries de sprints par semaine.Par série, il court 9 * 60 = «9*60=540»540 mètres.Par semaine, il court donc 540 * 9 = «540*9=4800»4800 mètres. ✗
`MathOctopus`	Jacques fait des séries de 3 sprints 3 fois par semaine.Chaque série est de 3 sprints, donc il court 3*3=«3*3=9»9 sprints par semaine.Chaque sprint est de 60 mètres, donc il court 60*9=«60*9=540»540 mètres par semaine. ✓
Answer	**540**
Generated Cases in Spanish	
Query	Los patos de Janet ponen 16 huevos por día. Ella come tres en el desayuno todas las mañanas y usa cuatro para hornear magdalenas para sus amigos todos los días. Vende lo que sobra en el mercado de productores diariamente a $2 el huevo fresco de pato. ¿Cuánto gana en dólares todos los días en el mercado de productores?
LLaMA 2	Janet comienza con 16 huevos frescos por día - 3 huevos frescos que come para desayuno = «16-3=13»13 huevos frescos que no se comen.Janet usa 13 huevos frescos - 4 huevos frescos para hornear magdalenas = «13-4=9»9 huevos frescos que no se comen.Janet vende 9 huevos frescos - 2 huevos frescos que se venden en el mercado de productores = «9-2=7»7 huevos frescos que se venden en el mercado de productores.Janet gana 7 huevos frescos * $2/huevo fresco = $«7*2=14»14 en el mercado de productores todos los días. ✗
`MathOctopus`	Janet come 3 huevos por día y usa 4 huevos por día, por lo que tiene 3 + 4 = «3+4=7»7 huevos que no usa para nada.Los patos de Janet ponen 16 huevos por día, por lo que tiene 16 - 7 = «16-7=9»9 huevos excedentes.Vende los 9 huevos excedentes por $2 cada uno, por lo que gana 9 * 2 = $«9*2=18»18 todos los días en el mercado de productores.✓
Answer	**18**

Table 12: Generated responses compared between LLaMA and ours within 7B-level on MGSM.