When natural language is not enough: The limits of in-context learning demonstrations in multilingual reasoning

Anonymous ACL submission

Abstract

Previous studies have demonstrated the effectiveness of *reasoning methods* in eliciting multi-step reasoned answers from Large Language Models (LLMs) by leveraging in-context demonstrations. These methods, exemplified by Chain-of-Thought (CoT) and Program-Aided Language Models (PAL), have been shown to perform well in monolingual contexts, primarily in English. There has, however, been limited exploration of their abilities in other languages. To gain a deeper understanding of the role of reasoning methods for in-context demonstrations, we propose a multidimensional analysis in languages beyond English, focusing on arithmetic and symbolic reasoning tasks. Our findings indicate that the effectiveness of reasoning methods varies significantly across different languages and models. Specifically, CoT, which relies on natural language demonstrations, tends to be more effective in high-resource than in low-resource languages. Conversely, the structured nature of PAL in-context demonstrations facilitates multilingual comprehension, enabling LLMs to generate programmatic answers in both high- and low-resource languages. This leads to significant improvements in the accuracy and quality of the generated responses.

1 Introduction

One of the emergent properties of Large Language Models (LLMs) is the ability to solve tasks through prompts defined by structured patterns. This phenomenon, known as in-context learning (Brown et al., 2020), allows a task to be solved without updating the model parameters by using only the input. In light of the success of in-context learning, there has been increased interest in better analyzing the factors that influence how it works, such as the selection of demonstrations (Liu et al., 2022; Rubin et al., 2022; Zhao et al., 2023) and prompt design (Zhang et al., 2022; Si et al., 2023).

In the case of *reasoning methods*, Chain-of-Thought (CoT) (Kojima et al., 2023; Wei et al., 2023), and Program-Aided Language Models (PAL) (Gao et al., 2022; Chen et al., 2023b) have emerged as two effective approaches. The first method, CoT, breaks down a reasoning problem into a series of intermediate steps using natural language, making it more general, flexible, and understandable. PAL offers reasoning solutions via Python functions, with its step-by-step programming code leading to more rigorous and structured reasoning.

While previous contributions have demonstrated the operation of in-context learning reasoning methods largely in English, a number of emerging works have investigated multilingual reasoning. Shi et al. (2022) have shown that the effectiveness of CoT rationales is limited to the languages most represented in the LLM's pre-training data. Hence, several studies addressed the problem by proposing prompting mechanisms that translate the problem to English (Huang et al., 2023), elicit cross-lingual alignments for enabling reasoning (Qin et al., 2023), or self-correction mechanisms (Ranaldi et al., 2024). However, limited attention is afforded to studying the type and role of in-context demonstrations and the effect of scale on multilingual reasoning with LLMs.

In this paper, we conduct an in-depth study to evaluate the role of reasoning methods beyond English. Taking previous work a step further, we study the operation of reasoning methods by analyzing the effects of different types of in-context learning demonstrations on multilingual generative reasoning abilities. We aim to identify different LLMs' characteristics, capabilities, and limitations in reasoning tasks performed in 26 languages.

This leads to the main research questions of this paper:

RQ1: What are the differences between natural language demonstrations and structured in-context

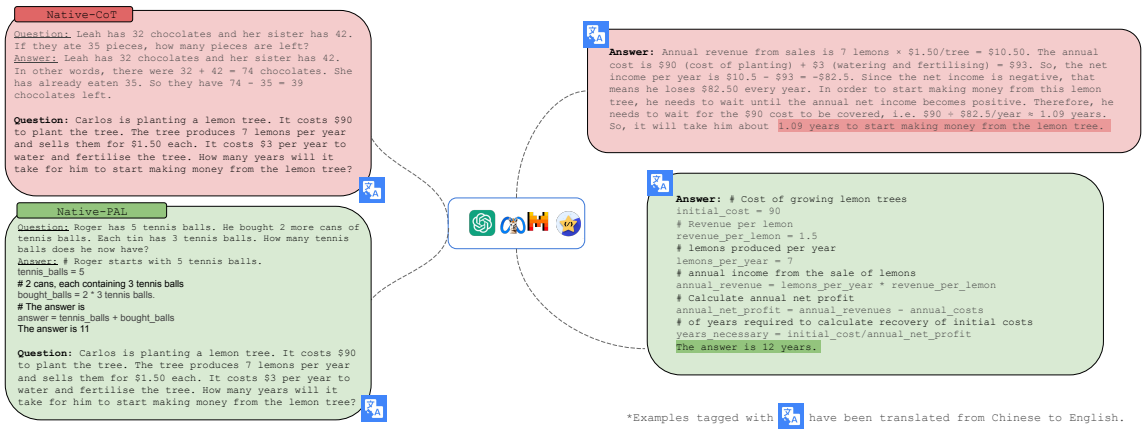

Figure 1: The different *reasoning methods* proposed in our analysis. We explore the impact of in-context demonstrations on multilingual tasks (Section 3.1) and the performances achieved by different LLMs (Section 3.2). *As indicated in the figure, we have translated two examples of prompts from Chinese to English to improve understandability.

demonstrations in multilingual reasoning?

RQ2: What are the impacts and limits of natural language in-context demonstrations beyond English?

RQ3: Do high and low-resource languages respond differently to reasoning methods?

We use two types of in-context demonstrations: `CoT` and `PAL` in zero and few-shot settings (as shown in Figure 1). For multilingual CoT, we use a series of natural language demonstrations either in English or in specific target languages following Shi et al. (2022). Similarly, for PAL, we propose amethod by extending the original in English (Gao et al., 2022) to additional languages.

We select a variety of multilingual reasoning tasks covering mathematical, commonsense reasoning, and natural language inference tasks. These tasks are MGSM (Shi et al., 2022) and MSVAMP (Chen et al., 2023a), which consist of mathematical reasoning problems, and XCOPA (Ponti et al., 2020), PAWS-X (Yang et al., 2019) and XLNI (Conneau et al., 2018) which consist of commonsense reasoning and natural language inference.

Finally, we select a range of different LLMs to explore the LLM family, LLM size, and purpose of construction for a comprehensive evaluation. Specifically, we employ GPTs (OpenAI, 2023) models for the results obtained in reasoning tasks, different versions of Llama2 (Touvron et al., 2023) and Mistral (Jiang et al., 2024) for the improvements achieved by smaller-scale versions; and finally, StarCoder (Li et al., 2023) and CodeLlama (Rozière et al., 2024) for the coding capabilities.

The main findings of our paper are:

- Reasoning methods are able to improve performance on non-English reasoning tasks. In fact, both CoT and PAL improve performance, although their effect on multilingual reasoning tasks varies greatly depending on the language and LLM.
- However, in the natural language in-context demonstrations used in the CoT, limitations can be seen in some languages. On the other hand, we observe that the structured reasoning of program demonstrations (i.e., PAL), are less ambiguous than natural language, and are more transferable between languages. PAL benefits from more structured reasoning, and this shows stronger performance for non-English tasks and, in particular, for low-resource languages.
- Finally, we show that LLMs are able to understand problems described in both low and high-resource language questions, even if performance is somewhat lower than in English. In addition, LLMs are able to generate reasoning in English even if the question is phrased in another language.

2 Reasoning Methods Beyond English

In-context reasoning methods are popular prompting strategies that elicit Large Language Models (LLMs) to generate multi-step reasoned answers as introduced in Section 2.1. Although these methods demonstrate their functionality in various tasks,

evaluations and further studies are primarily conducted in English, leaving other languages unexplored (Section 2.2). Hence, we propose a systematic study of the impact of reasoning methods in languages other than English (Section 2.3).

2.1 In-context Reasoning Methods

These methods, best represented by Chain-of-Thought (CoT) (Wei et al., 2023) and Program-Aided Language Models (PAL) (Gao et al., 2022), are popular prompting strategies that introduce in-context demonstrations. These examples elicit LLMs to solve complex problems by simplifying them and breaking them down into a series of sub-problems. The CoT-based methods operate in zero-shot (Kojima et al., 2023), few-shot (Wei et al., 2023), self-consistent way (Wang et al., 2023). In contrast, PAL uses a code interpreter (Zhou et al., 2023) or code-like structured demonstrations (Gao et al., 2022).

2.2 Reasoning Across Languages

Several earlier works have studied the performances of CoT prompting in different languages. Shi et al. (2022) tested the effectiveness of native in-context CoT that are manually translated rationales in a specific language (i.e., `Native-CoT` in Table 1) and English in-context CoT (i.e., `En-CoT`). En-CoT are composed of questions in the native language and rationales in English. Qin et al. (2023) inspired by (Huang et al., 2023) and (Wang et al., 2023), proposed two-step CoT prompting (see Table 7). Finally, Ranaldi et al. (2024) proposed a prompt-based self-correction strategy as described in Appendix B. However, these studies focused on demonstrating the performance of CoT and evaluated methods on large English-focussed LLMs. Thus, previous works left a gap in the study of the type of multilingual in-context demonstrations on both the impacts and effects they have on reasoning on different scales of LLMs.

`Native-CoT`

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。

Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

Table 1: Native Chain-of-Thought, as proposed in (Shi et al., 2022) (for simplicity, we have reduced the shot, but the original is 6-shot). The in-context question and the rationales are in the specific language (Chinese in this example). The version with English rationales is `En-CoT` as detailed in Appendix B).

`Native-PAL`

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A:
```
# 罗杰从5 个网球开始。
tennis_balls = 5
# 2 罐，每罐装3 个网球
bought_balls = 2 * 3 tennis balls.
# 答案是
answer = tennis_balls + bought_balls
# 答案是11
```

Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

Table 2: Native Program-Aided Language Models (`Native-PAL`) (we reported one-shot as in Table 1). The in-context questions and the demonstrations are in the native language (Chinese in this example). The `En-PAL` version have English commented answers as detailed in Appendix E).

2.3 Aligning Reasoning Methods

Inspired by previous work, we take the next step by proposing an in-depth evaluation of the effect of in-context demonstrations used in the *reasoning methods*. We conduct our analysis on different LLMs chosen by family, capabilities, and purpose of construction (Section 3.2) by using appropriate tasks presented in Section 3.1. Our contribution aims to study the effect of different types of in-context demonstrations in different languages by discussing their limitations and the functionality that these methods are able to supply.

Our experiments explore the following key points: (i) constructing multilingual evaluations by extending PAL (as introduced later) and applying multilingual CoT methods (Shi et al., 2022; Huang et al., 2023) on different models using carefully designed benchmarking tasks; (ii) conducting an investigation into the effects of in-context demonstrations across various languages; and (iii) analyzing the different impact of in-context reasoning approaches for high- and low-resource languages.

Multilingual PAL To extend multilingual evaluation to the PAL reasoning method, we propose

a specially constructed language-specific version (showed in Table 2) by transferring the prompts proposed in (Shi et al., 2022) into programs-like demonstrations as done in (Gao et al., 2022).

3 Experimental setup

3.1 Data

To study the impact of reasoning methods in multilingual tasks, we use MGSM (Shi et al., 2022), MSVAMP (Chen et al., 2023a), XNLI (Conneau et al., 2018), and PAWS-X (Yang et al., 2019), XCOPA (Ponti et al., 2020).

Understanding tasks To assess multilingual comprehension abilities, we use XNLI and PAWS-X. The first is an extension of Stanford Natural Language Inference data set (Bowman et al., 2015) to 15 languages and, based on premise and hypothesis, requires the model to determine whether the hypothesis is entailed, contradicted, or neutral in 15 different languages. The second, Paraphrase Adversaries from Word Scrambling (PAWS-X), contains two sentences and requires the model to judge whether they paraphrase each other in 7 languages.

Commonsense Reasoning task The Cross-lingual Choice of Plausible Alternatives (XCOPA) (Ponti et al., 2020) is based on one premise and two choices. It asks the model to choose which one is the result or cause of the premise, covering 11 languages from diverse families.

Arithmetic Reasoning task To evaluate the problem-solving abilities, we use the extension of GSM8K Cobbe et al. (2021) and SVAMP (Patel et al., 2021). Respectively, Multilingual Grade School Math (MGSM) (Shi et al., 2022) and Multilingual Simple Variations on Arithmetic Math word Problems (MSVAMP) (Chen et al., 2023a). In both original cases, the authors proposed a benchmark of mathematical problems in English. The examples have the following structure: a math word problem in natural language and a target answer in numbers. Shi et al. (2022); Chen et al. (2023a), in their contribution, selected a subset of instances from the official list of examples and translated them manually into 11 different languages, maintaining the structure of the input and output.

Evaluated Languages In our experiments, to promote open-source sharing, we use a list of tasks available in different languages; we provide detailed descriptions in Appendix A.

3.2 Models

We evaluate the effects of reasoning methods on different LLMs. Following previous work, we use three models from the GPT family; moreover, in additional experiments, we introduce other models from the Llama2 and Mistral families and StarCoder2. Hence, complementing previous evaluations, we choose models for (i) multilingual performances achieved by the GPTs and Llama2s (Ahuja et al., 2023),(ii) the monolingual abilities in mathematical reasoning achieved by Mixtral (Jiang et al., 2024) on GSM8K, and finally, (iii) the proficiency in coding for StarCoder2 (Li et al., 2023), CodeLlama (Rozière et al., 2024), and GPTinstruct (also for results in PAL (Gao et al., 2022; Ye et al., 2023)). We accessed the GPT models via the API, whereas we downloaded the other models from HuggingFace and ran inference locally. Appendix F and Appendix F describe the parameters and versions used in detail.

3.3 Prompting Methods

To conduct the study on robust models, we operate with state-of-the-art in-context learning methods and extend the experimental setting by introducing multilingual Program-Aided Language Models (PAL).

Arithmetic Reasoning Prompts We define two types of prompts for the MGSM and MSVAMP tasks by adapting CoT and PAL to multilingual scenarios. Hence, we use `En-CoT` and `Native-CoT` as in (Shi et al., 2022) (Table 1) and an adapted method proposed in (Qin et al., 2023; Huang et al., 2023) (see Appendix B). Concerning PAL, we introduce multilingual demonstrations as shown in Table 2 for `Native-PAL` and, to complete the settings `En-PAL` detailed in Appendix E.

Finally, to complete the experimental setting, we introduce `Cross-CoT` and `-PAL`. Both have the initial part as well as `Native-methods`, but unlike these, the models are elicited to deliver reasoned answers in English (detailed in Appendix W).

Understanding & Commonsense Prompts While we employ the workflow proposed in previous works for arithmetic tasks by performing experiments with zero and few-shot settings, for understanding and commonsense tasks, we define input templates that lead to the comprehension of LLMs and consequently aid generation. As described in detail in Appendix D, we construct prompts follow-

Model	Method	Mathematical		Understanding		Commonsense
		MGSM	MSVAMP	XNLI	PAWS-X	XCOPA
GPT-4	Direct	67.1	69.2	75.4	68.1	89.0
	CoT	68.4 (+1.3)	70.4 (+1.4)	**76.1** (+x.x)	**70.7** (+2.6)	**91.7** (+1.7)
	PAL	**71.2** (+4.1)	**71.7** (+2.5)	-	-	-
GPT-3.5-based	Direct	48.5	59.3	62.1	66.4	80.2
	CoT	55.9 (+6.4)	62.4 (+3.1)	**63.2** (+1.9)	**67.2** (+3.7)	**85.3** (+3.6)
	PAL	**57.5** (+9.0)	**63.9** (+4.3)	-	-	-
Llama-70-based	Direct	45.9	54.0	48.2	58.3	70.2
	CoT	51.0 (+5.1)	54.8 (+1.8)	**49.8** (+1.6)	**60.6** (+2.3)	**73.3** (+3.2)
	PAL	**51.5** (+5.6)	**55.7** (+1.7)	-	-	-
Llama-7-based	Direct	42.5	46.8	44.1	53.2	45.4
	CoT	46.1 (+3.6)	48.6 (+1.8)	**45.3** (+1.2)	**54.8** (+1.5)	**46.0** (+0.6)
	PAL	**47.2** (+4.7)	**49.4** (+2.6)	-	-	-
StarCoder2	Direct	41.6	46.8	-	-	-
	PAL	**45.1** (+3.5)	**48.6** (+1.8)	-	-	-
Mixtral8x7	Direct	**51.2**	56.2	42.5	57.6	**74.2**
	CoT	49.4 (-1.8)	**56.8** (+0.4)	**42.7** (+0.5)	**59.7** (+3.1)	72.7 (-1.5)
Mistral-7	Direct	**49.5**	**48.2**	38.5	56.3	**47.7**
	CoT	48.0 (-1.5)	47.8 (-0.4)	**40.1** (+1.6)	**58.4** (+2.1)	46.6 (-1.1)
	PAL	48.0 (-1.5)	48.0 (-0.2)	-	-	-

Table 3: The average accuracy scores achieved by models proposed in Section 3.2 using *reasoning methods* introduced in Section 3.3 (in **bold** the best performance per model and task). For `GPT-3.5-based`, we reported results achieved by `gpt-3.5-turbo` and `gpt-instruct`, and the same for `Llama-70-based` and `Llama-7-based`. Appendices I and J are reported detailed results.

ing (Ahuja et al., 2023) using the CoT prompting method to elicit multi-step generations.

Evaluation We evaluate performance using the accuracy score, following the approaches used in (Shi et al., 2022; Huang et al., 2023). Hence, we measure the exact match between generated outputs and labels[1] (Ahuja et al., 2023). We maintain the generation temperatures as recommended in the official papers.

4 Results

Large Language Models (LLMs) benefit from *reasoning methods* not merely in monolingual contexts (as amply demonstrated in English) but also in other languages. As discussed in Section 4.1, the in-context demonstrations beyond English elicit the LLMs to deliver multilingual multi-step reasoned answers; however, the operation differs depending on the type of method.

Although in-context demonstrations lead the models to generate more robust answers, bringing tangible improvements in multilingual tasks, the operation of these techniques differs depending on the model. As analyzed in Section 4.2, in-context rationales in natural language have a limited effect in some languages. On the other hand, structured program-of-thoughts demonstrations lead the models to stable generations. However, the impact of demonstrations varies according to the quality or quantity of rationales and the scale of model parameters (Section 4.3).

Finally, in Section 4.4, we examine the effects of high-resource languages on the final performance by discerning the factors that influence the generation of the final response and highlighting the matter of native language demonstrations in low-resource settings.

4.1 Reasoning Methods operate across languages

In-context reasoning methods empower the LLMs' multilingual performances in mathematical, commonsense reasoning and understanding tasks. Table 3 shows the differences in terms of performance between the language-adapted reasoning methods, i.e., `Native-CoT` and `Native-PAL`, and the baselines (i.e., `Direct`). The use of in-context multilingual demonstrations also brings clear benefits beyond English.

In particular, the results achieved by GPT-4 and GPT-3.5-based (GPT-3.5 and GPT-instruct)

[1]We extract target labels from the generated answers using regular expressions before calculating the exact match. In addition, for each task, we use *Instruction Templates* (Appendix G) to guide the model to stable generations and facilitate evaluation.

show a clear distinction between Native-CoT, Native-PAL and the baseline Direct.Also, Llama-70-based (Llama2-70 and Codellama-70) models obtain noticeable benefits from Native-CoT and Native-PAL prompting (complete results in Appendix I). Although these LLMs benefit the most from introducing reasoning methods in the prompting stage, further improvements are observable even in LLMs with fewer parameters. In particular, Llama2-7-based and StarCoder2 outperform the baselines when reasoning methods are used (see the average scores in Table 3). In contrast to the general trend, the models of the Mistral family do not perform as well on reasoning methods as the other models on mathematical tasks (see differences with reference averages in Table 3. These results demonstrate the benefit of multilingual in-context prompting not only in the mathematical tasks but also in natural language understanding, average accuracies in last three columns of Table 3 (details for each language in Tables 20 and 21 and commonsense task in Table 22).

However, although the averages are mainly positive, some phenomena emerge, such as negative differences (the baseline Direct outperforms the reasoning method) and, in the mathematical reasoning tasks, a disparity between CoT and PAL.

Specifically, PAL outperforms CoT consistently with accuracy averaging ranging from 1.5 up to 3 points (green values in Table 3). Hence, to gain a thorough understanding of the dynamics that emerge, we now explore how the structure of in-context demonstrations affects the generations provided by the models.

4.2 The Limits of Natural Language

The effect of the reasoning method relies on the solution strategy. Structured in-context demonstrations in a program-like manner (PAL) are more effective than natural language rationales(CoT) in multilingual mathematical reasoning tasks. Figures 2 display the differences between PAL and CoT using both in-context learning adapted to the specific language (Native-PAL and Native-CoT). Furthermore, to complete the analysis, the same experiments were performed with in-context demonstrations in English (En-PAL and En-CoT[2]).

In particular, in mathematical tasks, PAL outperforms CoT in eight languages over ten on average in the case of language-specific demonstrations (Native-CoT and Native-PAL). Furthermore, similar results emerged using the in-context in English as proposed in (Shi et al., 2022). Examining the results between languages shows that better improvements are obtained in low-resource languages (low- and high-resource[3], echoing previous work (Shi et al., 2022)). In particular, PAL consistently outperforms CoT in high-resource languages by about 1.8 average points while by more than 2.1 points in low-resource languages. The phenomenon is more marked in models beyond GPT-based (Llama2-based and StarCoder2).

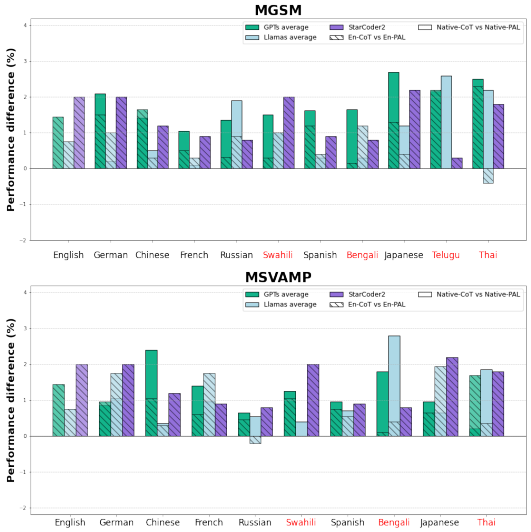

Figure 2: Performance difference between Native-PAL and Native-CoT for each individual language (low-resource languages in red) in MGSM and MSVAMP (we also reported the difference between En-PAL and En-CoT hatched). In Table 28 are reported the extended differences for all models.

Since the natural language of in-context rationales does not provide the same benefits as PAL, we examined the generations delivered by the different LLMs in detail to investigate the origin of the differences.

The structure of the Rationales The in-context demonstrations in natural language have the same structure as the previously proposed contribution, but they have different effects due to imbalances related to pre-training languages. Even though the Native-CoT consists of questions and demonstrations in a specific language, the generations are not always in the same language (Figure 3). Analyzing the composition of languages through the

[2]details on the structure of the prompts in Section 3.3

[3]high (German, Chinese, French, Russian, Spanish, Japanese) and low (Telugu, Bengali, Swali, Thai)

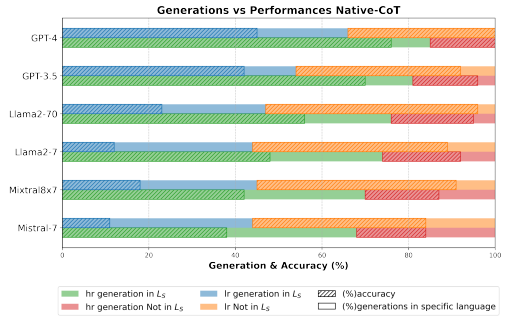

Figure 3: Percentage (%) of answers generated in specific language (L_S) and the relative accuracies for MGSM using `Native-CoT`. We reported averages for low-resources (lr) and high-resources (hr) languages.

framework OpenLID (Burchell et al., 2023), a difference emerges in the generations delivered for high-resource languages that are predominantly in the native language as opposed to low-resource that are predominantly in English despite the in-context in the specific language (detailed in Appendix S).

Hence, these generations impact the performances. The results in Figure 3 show that non-language-specific answers tend to be more accurate than other answers. In addition to the languages generated, a relationship emerges between performance and the average number of steps required to get correct answers. The number of $Hops$, i.e., the steps to reach the final solution, represented by natural language sentences, appears different between English and non-English (Figure 4). In contrast, PAL generations are less skewed between English and specific language answers (detailed results for each model in Appendix S).

4.3 The Role of Demonstrations

In-context demonstrations play a key role in complex multilingual scenarios because they promote reasoning in specific languages beyond English, as discussed in Sections 4.1. We investigated the

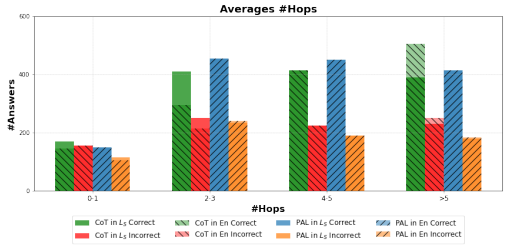

Figure 4: Average number of $Hops$ in answers delivered in specific language (L_S) and in English (En) (in Appendix S are reported for each model)

performance trend as in-context demonstrations increased by varying the quality and quantity of demonstrations operated. We repeated the previous experiments focusing on a mathematical task (MGSM), starting with zero- and increasing to 6-shots. The results (Figure 5) show that the positive impact of in-context demonstrations across the languages is related to the quality (as discussed in Section 4.2) and quantity of demonstrations used.

Quantitative Impacts The amount of in-context demonstrations is relevant (Figure 5). However, a distinction emerges between models and the number of de facto useful demonstrations. GPT-based models with 4-shots achieve results comparable to 6-shots (average accuracies in Figure 5). This balance does not occur in Llama2-70, CodeLlama-70, and Mixtral, which underperform as in-context demonstrations increase (see details in Figure 6). Finally, the smaller models (Llama2-7, Mistral-7, CodeLlama-7 StarCoder2) have conspicuous improvements as the number of demonstrations increases. However, there are divergences related to the languages, as discussed in Section 4.4.

Model		Δ MGSM	Δ MSVAMP	Δ XNLI	Δ PAWS-X	Δ XCOPA
`GPT-4`	**CoT**	+11.2	+3.2	+1.8	+3.6	+5.5
	PAL	+5.8	+2.6	-	-	-
`GPT-3.5`	**CoT**	+7.8	+6.1	+2.6	+2.8	+0.6
	PAL	+8.4	+5.4	-	-	-
`Llama-70`	**CoT**	+5.0	+4.8	-0.2	-0.4	-0.3
	PAL	+4.7	+4.6	-	-	-
`Llama-7`	**CoT**	+2.7	+0.8	+0.3	-1.2	+0.2
	PAL	+2.1	+0.2	-	-	-
`Mixtral`	**CoT**	+0.1	-0.4	-0.6	-0.5	+0.9
	PAL	-	-	-	-	-
`Mistral-7`	**CoT**	+0.2	-1.6	0.8	-0.7	+0.7
	PAL	-0.2	-1.2	-	-	-
`Starcoder2`	**PAL**	+3.7	+0.8	-	-	-

Table 4: Differences in term of accuracies (Δ) between `Cross-CoT` and `Cross-PAL` and the `Native-based` versions.

4.4 The Language of Reasoning Matters

Multilingual in-context demonstrations may benefit LLMs in applying solution strategies during generation; however, the language used to solve a given problem matters. By prompting in a specific language and eliciting the LLMs to generate reasoning in English (defined as `Cross-method`), we observed significant improvements in accuracy

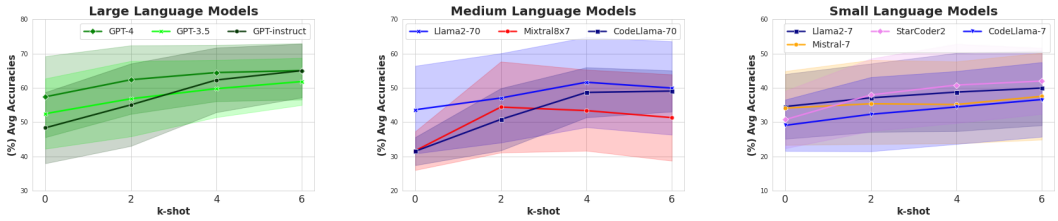

Figure 5: Average accuracies for all languages on MGSM using methods prompting in (Section 3.3) setting providing in input `k-shot` demonstrations with `k` equal to `{0,2,4,6}`. In Appendix M and Appendix O, detailed results are reported.

(see Table 4).

Specifically complementing previous work, we used two strategies: (1) delivered in-context demonstrations of reasoning answers in English (`En-CoT`, `En-PAL`). (2) delivered in-context demonstrations of reasoning answers in the native language and then elicited the model to reason and provide a solution in English, i.e., `Cross-method` (reported in Appendices W) As shown in Table 4, in both cases, the reasoning methods provided tangible benefits both in PAL and CoT. These latter results emphasized the LLMs' understanding and production abilities as reported in generations in Tables 31, 32. Finally, in the additional experiments in Appendix T, we show that our analysis can be transferred to further LLMs, and the observations discussed also emerge in these latter models.

5 Related Work

Large Language Models (LLMs) demonstrate in-context learning abilities (Min et al., 2022; Dong et al., 2023) to guide LLMs in generating desired task responses, marking the advent of the prompting era and surpassing the age of the intermediate steps and structured reasoning (Roy and Roth, 2015; Ling et al., 2017). Early works challenged the efficacy of in-context learning approaches to improve downstream performances. Gao et al. (2022) adapted the Chain-of-Thought (CoT) (Wei et al., 2023) approach by considering the proficiency of LLMs in producing code, proposing Program-Aided Language Models (PAL).

These approaches, called *reasoning methods*, demonstrated success, but the findings are limited to a single language (i.e., English). Hence, Shi et al. (2022) proposed a multilingual evaluation that Huang et al. (2023); Qin et al. (2023); Ranaldi et al. (2024) (see summary in Table 29) extended into cross-lingual by proposing a prompt mechanism to handle requests in any language and generate English CoT. While Ranaldi et al. (2024) proposed a single single-phase prompt, Huang et al. (2023) used a double-step mechanism reinforced with the self-consistency approach (Wang et al., 2023) as in (Qin et al., 2023). Although the proposed approaches achieve robust results, the role, effects, and limitations of multilingual in-context demonstrations on reasoning abilities remained under-explored.

This work proposes an analysis to evaluate LLMs' multilingual reasoning abilities. We investigate the impact that *reasoning methods* cause on final performance by studying the role and the limits of in-context demonstrations in different languages. The cornerstones can be outlined by the following points: (i) Analysis of the impact of multilingual reasoning methods on different tasks using several LLMs (selected by features and scope of construction); (ii) Examination of the limitations of multilingual in-context demonstrations in natural language and comparison with PAL; (iii) Study the role of in-context demonstrations by discerning between low- and high-resource languages.

6 Conclusion

The benefits of *reasoning methods* emerge beyond the English language. Our analysis shows that appropriately elicited LLMs are able to deliver structured answers in different languages. Indeed, by operating via CoT and PAL, we revealed that in-context demonstrations play a strategic role in improving performance in direct proportion to their quality and quantity. Our research highlights the need for a customized approach to employing reasoning methods for LLMs in different languages. It supports the demand for a reasonable combination of model scale, reasoning technique, and strategic use of in-context demonstrations to elicit the prospect of LLMs in different language landscapes.

Limitations

Due to the limitations imposed by the evaluation benchmarks and the cost of the OpenAI API, we conducted tests on five tasks and 26 languages in total, which only scratches the surface of the world's vast array of languages. In addition, our approaches are based on a single-stage prompting approach in English. It should be evaluated Self-consistency prompts (Wang et al., 2023) and using different configurations of cross-lingual in-context demonstrations. Finally, we tested the effectiveness of our method on GPT-based models (closed-source) and several models (open-source). In the future, it will be appropriate to study the generality of our model compared to other closed-source Large Language Models.

Finally, although we have considered and analyzed different versions distributed over 22 models in our work, we would like to take a closer look at the performance achieved by language-specific pre-trained models (better known as language-centered). However, at the moment, there are not many open resources comparable in size to those we have analyzed. In the future, we hope these models can be readily available to investigate this phenomenon better.

Ethics Statemet

In our work, ethical topics were not addressed. The data comes from open-source benchmarks, and statistics on language differences in commonly used pre-training data were obtained from official sources without touching on gender, sex, or race differences.

References

Kabir Ahuja, Harshita Diddee, Rishav Hada, Millicent Ochieng, Krithika Ramesh, Prachi Jain, Akshay Nambi, Tanuja Ganu, Sameer Segal, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2023. MEGA: Multilingual evaluation of generative AI. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4232–4267, Singapore. Association for Computational Linguistics.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

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.

Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879, Toronto, Canada. Association for Computational Linguistics.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Yangqiu Song, Dongmei Zhang, and Jia Li. 2023a. Breaking language barriers in multilingual mathematical reasoning: Insights and observations.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023b. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks.

Jiale Cheng, Sahand Sabour, Hao Sun, Zhuang Chen, and Minlie Huang. 2023. PAL: Persona-augmented emotional support conversation generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 535–554, Toronto, Canada. Association for Computational Linguistics.

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.

Common Crawl. 2021. Common crawl 2021. Web. Accessed: 2023-12-12.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2022. Pal: Program-aided language models. *arXiv preprint arXiv:2211.10435*.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nour Fahmy, Urvashi Bhattacharyya, Wenhao Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jan Ebert, Tri Dao, Mayank Mishra, Alex Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you!

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

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.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, Elena Sofia Ruzzetti, and Fabio Massimo Zanzotto. 2024. Empowering multi-step reasoning across languages via tree-of-thoughts.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2024. Code llama: Open foundation models for code.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners.

Chenglei Si, Zhe Gan, Zhengyuan Yang, Shuohang Wang, Jianfeng Wang, Jordan Boyd-Graber, and Lijuan Wang. 2023. Prompting gpt-3 to be reliable.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology.

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. 2023. Llama 2: Open foundation and fine-tuned chat models.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Xi Ye, Qiaochu Chen, Isil Dillig, and Greg Durrett. 2023. Satlm: Satisfiability-aided language models using declarative prompting.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning.

James Zhao, Yuxi Xie, Kenji Kawaguchi, Junxian He, and Michael Xie. 2023. Automatic model selection with large language models for reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 758–783, Singapore. Association for Computational Linguistics.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification.

A Proposed Task

Dataset	Task	Languages	#Languages
MGSM	mathematical reasoning	Bengali (bn), Chinese (zh), French (fr), Thai (th) German (de), Japanese (jp), Russian (ru), Telugu (te) Spanish (es), Swahili (sw)	10
MSVAMP	mathematical reasoning	Bengali (be), Chinese (zh), French (fr), Thai (th) German (de), Japanese (jp), Russian (ru) Spanish (es), Swahili (sw)	9
XNLI	natural language inference	English (en), German (de), Russian (ru), French (fr), Spanish (es), Chinese (zh), Vietnamese (vi), Turkish (tr), Arabic (ar), Greek (el), Thai (th), Bulgarian (bg), Urdu (ur), Swahili (sw), Hindi (hi)	15
XCOPA	commonsense reasoning	Chinese (zh), Italian (it), Vietnamese (vi), Indonesian (in), Turkish (tr), Thai (th), Estonian (et), Tamil (ta), Swahili (sw), Haitian (ht), Quechua (qu)	11
PAWS-X	paraphrase identification	English (en), German (de), Japanese (jp), French (fr), Spanish (es), Chinese (zh), Korean (ko)	7

Table 5: Languages present in datasets used in this work.

M K-shot per Model

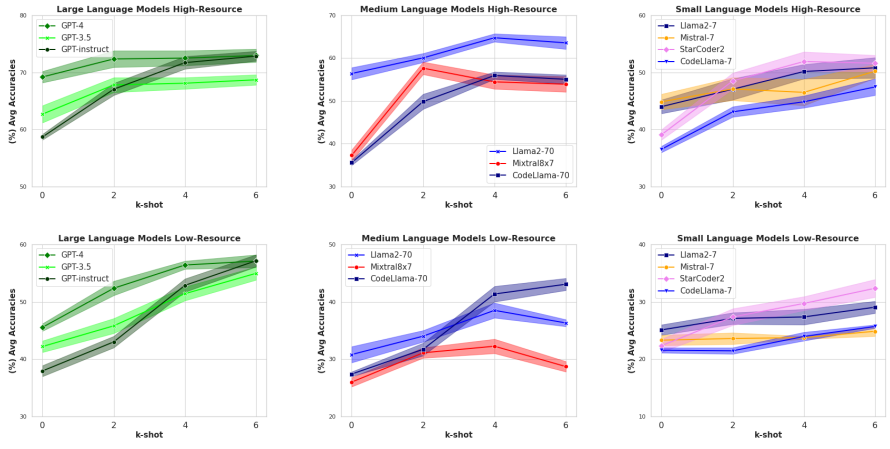

Figure 6: Average accuracies on mathematical reasoning task (MGSM) using methods proposed in (Section 3.3) setting providing in input `k-shot` demonstrations with `k` equal to `{0,2,4,6}`. In Appendix M and Appendix O, detailed results are reported.

O K-shot per Language using CoT

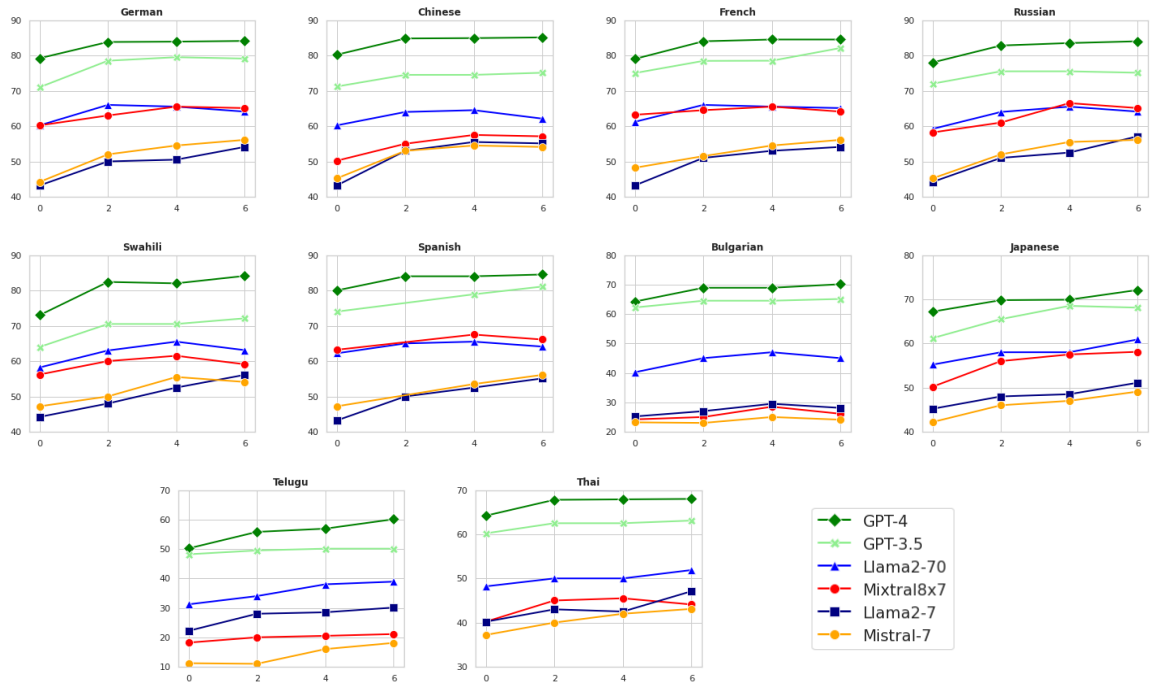

Figure 7: Acciracies (%) on MGSM using `Native-CoT` (Section 3.3) setting providing in input `k-shot` demonstrations with `k` equal to `{0,2,4,6}`.

P K-shot per Language using PAL

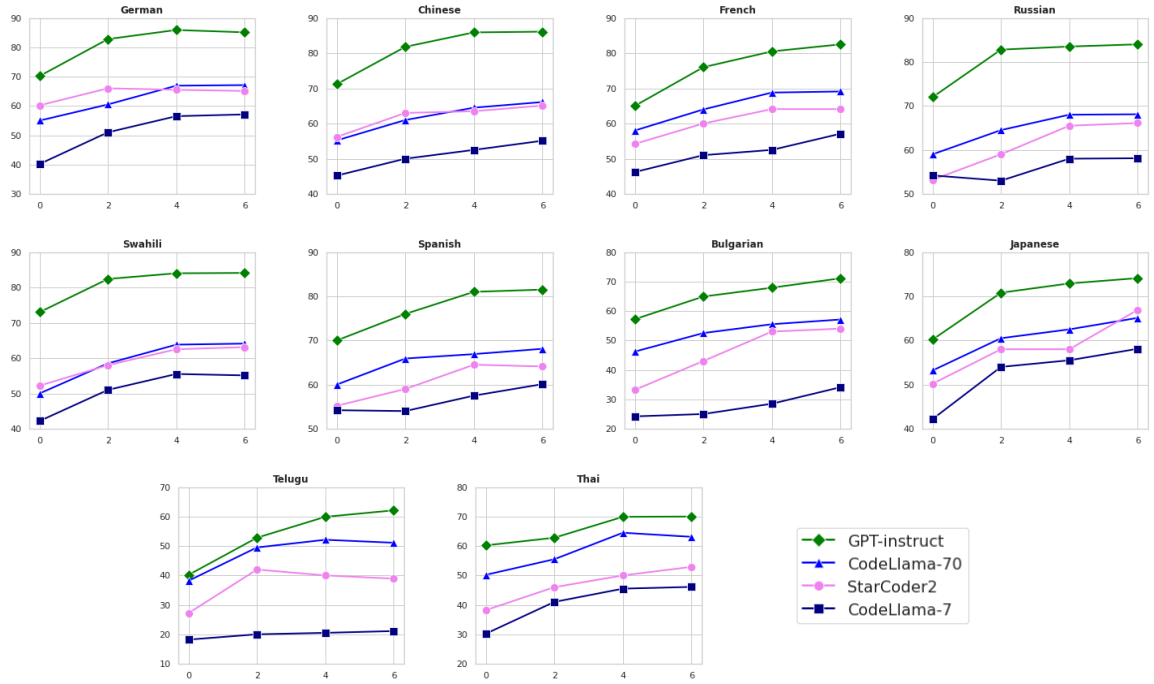

Figure 8: Acciracies (%) on MGSM using `Native-PAL` (Section 3.3) setting providing in input `k-shot` demonstrations with `k` equal to `{0,2,4,6}`.

B State-of-art Prompting Methods

Direct (Question in Chinese without CoT)

Q: ： 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 11
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

Native-CoT (Question and CoT Answer in Chinese)

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

En-CoT (Question in Chinese and CoT Answer in English)

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.
Q: 利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
A:

Table 6: Chain-of-Thought as proposed in (Shi et al., 2022) (for simplicity we have reduced the shot but the original is 6-shot). Given a problem in specific language, the following prompts are `Direct`, `Native-CoT` (without additional languages) and `En-CoT`, the original question in specific language with answers in English.

CLIP First-Step

Please act as an expert in multi-lingual understanding in [Specific Language L_s].
Question: [Given sentence X in L_s]
Let's understand the task in [Target Language L_t] step-by-step!

CLIP Second-Step

After understanding, you should act as an expert in mathematics in [Language L_t].
Let's resolve the task you understand above step-by-step!

Table 7: CLIP (Qin et al., 2023) where the prompt is split into two phases: there is the alignment of the different languages, and then, there is the solving mechanism for the specific language.

Cross-ToT

Simulate the collaboration of $\{n\}$ mathematicians answering a question in their mother tongue: L_1, L_2, ... and L_n. They all start Step1 from a separate thought process, step by step, each explaining their thought process. Following Step1, each expert refines and develops their thought process by comparing themselves with others. This process continues until a definitive answer to the question is obtained.
Question: [Question in Language L_1]
Answer: [num].

Table 8: `Cross-ToT` prompting (Ranaldi et al., 2024) that using Tree-of-Thoughts method elicit the model to produce multi-step reasoning processes in different languages.

C Prompting Methods Arithmetic Reasoning Tasks

In this work, as introduced in Section 3, we propose the Cross-lingual extension of Program-Aided Language Models (Cross-PAL) as shown in Table 34 (detailed in Appendix E), and a Cross-lingual version of CoT as shown in Table 33. In detail, in both settings, the prompt is a few-shots as proposed in (Wei et al., 2023) for CoT and in (Gao et al., 2022) for PAL, respectively; however, unlike the previous versions, the question-answer pairs (the answers are a CoT demonstration) are proposed in the languages evaluated in each task. Moreover, we use additional configurations as proposed by Shi et al. (2022): `"Direct"` prompt, i.e., question and answer in the original language; the `"Native-CoT"` prompt, i.e., question and answer CoT in the original language; the `"En-CoT"` prompt specific language question and answer CoT in English (see prompts in Appendix B). Furthermore, in order to analyse the effect of reducing the in-context examples down to zero-shots we propose additional settings esemplifing the number and the typology of demonstrations that compose the prompt.

D Prompts for Understanting & Commonsense Reasoning Tasks

As far as prompts for natural language understanding and commonsense reasoning tasks are concerned, we follow the methods proposed by state-of-the-art works. Hence, following Ahuja et al. (2023), to construct prompts that lead Large Language Models (LLMs) to produce stable and structured answers, we define a sequence consisting of *Task Instruction*, *Demonstration*, and *Task Problem*. In particular, the *Task Instruction* is the initial instruction that defines the type of task and the desired answer. Then, there is a body composed of Demonstrations that are related to the number of shots. For example, in the few-shot settings such as CoT proposed in (Shi et al., 2022), the demonstrations are composed of questions and desired outputs. Finally, the final part consists of questions about the tasks we are analyzing. As in Appendix B, we propose `Direct`, `En-CoT` and `Cross-CoT` configurations while we do not use PAL as it is not suitable for this type of task. In Table 14, we report the selected templates. Table 30, 31 and 32 report the demonstrations, input and outputs generated.

E Program-Aided Language Models Prompts

In this paper, as introduced in Section 3.3, we propose a novel Cross-lingual extension of the Program-Aided Language Models (Gao et al., 2022) (Cross-PAL) method. The following tables show the prompts used for the final evaluation.

Program-Aided Language Models (PAL)

```
Q: Roger has 5 tennis balls. He buys 2 more
cans of tennis balls. Each can has 3 tennis
balls. How many tennis balls does he have now?
A: Roger started with 5 tennis balls.
   tennis_balls = 5
   2 cans of 3 tennis balls each is
   bought_balls = 2 * 3 tennis balls.
   The answer is
   answer = tennis_balls + bought_balls
   The answer is 11
Q: Kyle bought last year's best-selling book
for $19.50. This is with a 25% discount from the
original price. What was the original price?
A:
```

Table 9: This is an example prompt of the PAL method proposed by (Gao et al., 2022).

En-PAL

```
Q: Roger has 5 tennis balls. He buys 2 more
cans of tennis balls. Each can has 3 tennis
balls. How many tennis balls does he have now?
A: Roger started with 5 tennis balls.
   tennis_balls = 5
   2 cans of 3 tennis balls each is
   bought_balls = 2 * 3 tennis balls.
   The answer is
   answer = tennis_balls + bought_balls
   The answer is 11
Q: Kylar geht ins Kaufhaus, um Gläser für seine
neue Wohnung zu erwerben. Ein Glas kostet 5
US-Dollar, aber jedes weitere Glas kostet nur
60% des Ausgangspreises. Kylar möchte 16 Gläser
kaufen. Wie viel muss er dafür ausgeben?
A:
```

Table 10: In En-PAL we use the same setting proposed in Table 6 but in contrast to En-PAL we use PAL demonstrations.

Cross **Program-Aided Language Models**

```
Q: Michael hat 58 Golfbälle. Am Dienstag hat
er 23 Golfbälle verloren. Am Mittwoch hat er
2 weitere verloren. Wie viele Golfbälle hat er
Mittwoch am Ende des Tages?
A: Michael hat 58 Golfbälle.
   initial = 58
   Am Dienstag verlor er 23 Golfbälle
   lost_tuesday = 23
   Am Mittwoch verlor er 2 Golfbälle
   lost_wednesday = 2
   Golfbälle abzüglich der verlorenen
    answer = initial - lost_tuesday
     - lost_wednesday
    Die Antwort ist 33
*(final question as in Table 10)
```

Table 11: In Cross-PAL, we use the same setting proposed in Table 10 but in contrast to En-PAL, we use PAL demonstrations in the same language of the question.

F Model and Hyperparameters

In our experimental setting, as introduced in Section 3.2, we propose different LLMs: (i) three models from the GPT family (OpenAI, 2023): GPT-3.5 (`gpt-3.5-turbo-0125`), Codex (`gpt-3.5-turbo-instruct`) and GPT-4 (`gpt-4`); (ii) four models from the Llama-2 family (Touvron et al., 2023): Llama2-7b, Llama2-70b, CodeLlama-7 and CodeLlama-70; (iii) two models of the MistralAI family: Mistral-7b and Mixtral (Jiang et al., 2024); (iv) finally, StarCoder2-15b (Li et al., 2023). In particular, GPTs models are used via API, while for the others, we used versions of the quantized to 4-bit models that use GPTQ (see detailed versions in Table 27)

Furthermore, we have added additional LLMs in the additional experiments presented in the Appendix T. These models are two from Orca2 (Mukherjee et al., 2023), two from Yi (AI et al., 2024), two models of the Google (Team et al., 2024), three from Wizard (Luo et al., 2023), and three from Tulu (Ivison et al., 2023) families.

As discussed in the limitations, our choices are related to reproducibility and the cost associated with non-open-source models. We use closed-source API and the 4-bit GPTQ quantized version of the model on four 48GB NVIDIA RTXA600 GPUs for all experiments performed only in inference.

Finally, the generation temperature used varies from $\tau = 0$ of GPT models to $\tau = 0.5$ of Llama2s. We choose these temperatures for (mostly) deterministic outputs, with a maximum token length of 256. The other parameters are left unchanged as recommended by the official resources. We will release the code and the dataset upon acceptance of the paper.

Language	Percentage
English (en)	46.3%
Russian (ru)	6.0%
German (de)	5.4%
Chinese (zh)	5.3%
French (fr)	4.4%
Japanese (ja)	4.3%
Spanish (es)	4.2%
Other	23.1%

Table 12: Language distribution of CommonCrawl (Common Crawl, 2021).

G Instruction Template for MGSM and MSVAMP

This section contains the *Instruction Templates* used during the evaluation phase. The following templates have been specially constructed to simplify the evaluation and route the generation of the analyzed models.

Instruction Template for:

Direct
(few-shot examples as showed for Direct *in Table 6)* **Q:** *[question in evaluated language]* For clarity, the answer should have the following format:'Answer:[num]'. (**N.B.** translated in {evaluated language})
Native-CoT
(few-shot examples as showed for Native-CoT *in Table 6)* **Q:** *[question in evaluated language]* Let's think step by step! For clarity, the answer should have the following format:'Answer:[num]'. (**N.B.** translated in {evaluated language})
En-CoT
(few-shot examples as showed for En-CoT *in Table 6)* **Q:** *[question in evaluated language]* Let's think step by step! For clarity, the answer should have the following format:'Answer:[num]'.
Cross-CoT
Given the following examples, please act as an expert in multilingual understanding in {**evaluated language**}. *(few-shot examples as in* Native-CoT *in Table 6), but the final instructions are in English)* **Q:** *[question in evaluated language]* After understanding, act as an expert in arithmetic reasoning in English. Let's answer the question step-by-step! For clarity, the answer should have the following format:'Answer:[num]'.
PAL & **En-PAL**
(few-shot examples as Table 9 for PAL *and Table 10 for* En-PAL*)* **Q:** *[question in evaluated language]* After understanding you act as a programmer by writing the solution. For clarity, the answer should have the following format:'The answer is [num]'.
Cross-PAL
Given the following examples, please act as an expert in multilingual understanding in {**evaluated language**}. *(few-shot examples as showed in Table 11)* **Q:** *[question in evaluated language]* After understanding answer the question, you should act as a programmer in English. For clarity, the answer should have the following format: 'The answer is [num]'.

Table 13: *Instruction Template* for Cross-CoT and Cross-PAL. The structure is defined by a set of in-context examples (zero examples, in the 0-shot case), the question in {**evaluated language**}, the final instruction part and a special template to guide generation and support the final evaluation.

H Task Instruction for XNLI, XCOPA ans PAWS-X

Task Instruction for:

XNLI
You are an NLP assistant whose purpose is to solve Natural Language Inference (NLI) problems in {**evaluated language**}. NLI is the task of determining the inference relation between two (short, ordered) texts: entailment, contradiction, or neutral. Answer as concisely as possible in the same format as the examples below:
XCOPA
You are an AI assistant whose purpose is to perform open-domain commonsense causal reasoning in {**evaluated language**}. You will be provided a premise and two alternatives, where the task is to select the alternative that more plausibly has a causal relation with the premise. Answer as concisely as possible in the same format as the examples below:
PAWS-X
You are an NLP assistant whose purpose is to perform Paraphrase Identification in {**evaluated language**}.The goal of Paraphrase Identification is to determine whether a pair of sentences have the same meaning. Answer as concisely as possible in the same format as the examples below:

Table 14: *Task Instruction for* XNLI, XCOPA and PAWS-X as proposed in (Ahuja et al., 2023). List of the Basic Prompt is in Table 15

Benchmark	**#Test**	**Final Prompt**
MGSM	250	Q: {problem}
MSVAMP	1000	Q: {problem}
XCOPA	200	Here is a premise: {premise}. What is the {question}? Help me pick the more plausible option: -choice1: {choice1}, -choice2: {choice2}
XNLI	200	{premise}. Based on the previous passage, is it true that {hypothesis}? Yes, No, or Maybe?
PAWS-X	200	Sentence 1: {sentence1} Sentence 2: {sentence2} Question: Does Sentence 1 paraphrase Sentence 2? Yes or No?

Table 15: The prompt of each task (excepted for MGSM and MSVAMP) that is systematically added following the instructions we defined in Table 14. The column **#Test** denotes the number of instances for each language in the test set proposed by the authors. The constructions of these tasks are derived from translations (manual or automatic) of subsets of the original monolingual versions (in English) as explained in Section 3.1.

I Results Arithmetic Reasoning Tasks Larger Models

The following evaluations were made by prompting the models presented in Section 3.2 with the methods presented in Section 3.3 (see Appendix B and Appendix E where the prompts are described in detail).

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
GPT-3.5	Direct	56.0	60.0	62.0	62.0	48.0	61.2	33.6	52.8	7.6	42.2	48.5
	Native-CoT	70.0	60.6	64.2	62.4	52.4	63.2	40.4	59.4	39.2	46.6	55.9
	En-CoT	71.8	63.2	70.0	65.6	55.2	69.6	50.4	60.6	40.0	48.0	59.0
	Cross-CoT	75.2	72.2	74.0	**72.8**	**66.2**	**72.6**	**63.8**	64.6	46.2	58.8	67.0
	Native-PAL	70.8	61.2	64.6	63.8	53.2	63.8	42.6	59.6	41.0	52.8	57.3
	En-PAL	72.0	65.0	70.2	64.6	54.8	70.2	49.8	61.8	41.4	53.2	60.4
	Cross-PAL	**77.0**	**73.4**	**76.2**	68.8	65.2	70.8	63.6	**69.8**	53.0	**64.4**	**68.2**
$GPT_{instruct}$	Native-PAL	71.0	62.0	63.8	64.0	53.6	64.0	42.6	60.8	42.0	53.8	57.8
	En-PAL	71.8	65.8	70.2	65.0	55.0	69.8	50.6	61.2	41.0	58.6	60.8
	Cross-PAL	**78.0**	**75.8**	76.6	**70.2**	65.8	70.8	63.4	**66.8**	**53.6**	64.0	**68.5**
Llama2-70	Direct	52.2	55.0	58.2	60.0	46.4	58.6	30.2	48.6	9.2	41.0	45.9
	Native-CoT	63.8	60.4	60.2	58.2	51.4	61.4	28.8	50.6	28.4	44.2	51.0
	En-CoT	64.0	61.4	61.6	61.4	50.6	62.8	33.8	54.2	35.8	49.0	54.0
	Cross-CoT	64.8	62.6	**64.8**	**64.6**	53.4	**64.0**	**41.8**	56.4	**36.8**	**51.2**	56.0
CodeLlama-70	Native-PAL	64.0	60.6	59.8	60.0	52.2	60.8	29.0	51.2	31.0	46.8	51.5
	En-PAL	65.0	62.2	61.8	61.8	52.6	61.6	34.6	55.4	33.8	47.4	53.0
	Cross-PAL	**65.8**	**63.6**	62.6	64.2	**54.2**	63.8	41.6	**57.4**	36.6	51.4	**56.2**
Mixtral8x7	Direct	**58.2**	**62.4**	**64.4**	**62.8**	**54.2**	**62.8**	**35.0**	54.2	12.8	44.6	51.2
	Native-CoT	56.8	58.2	57.6	56.8	50.2	62.0	30.6	55.6	18.6	45.4	49.4
	En-CoT	55.8	59.4	58.6	58.4	51.0	63.0	44.8	56.8	**22.2**	**46.6**	**51.6**
	Cross-CoT	57.6	56.8	58.2	57.2	53.0	61.2	28.4	**58.6**	20.0	45.2	49.5

Table 16: Accuracies (%) on MGSM using the reasoning methods described in Appendix C (for each model, we reported best performances per language and per method in **bold**).

Model	Method	de	zh	fr	ru	sw	es	bn	ja	th	Avg
GPT-3.5	Direct	60.3	66.2	63.5	60.3	59.2	69.2	9.6	68.9	36.2	59.3
	Native-CoT	68.9	76.5	77.8	68.5	66.3	74.5	12.1	73.1	43.5	62.4
	En-CoT	73.9	78.4	78.2	70.9	68.4	74.6	14.4	74.0	46.1	64.3
	Cross-CoT	78.4	**78.6**	79.3	74.8	70.4	75.2	**41.0**	**76.2**	51.4	69.4
	Native-PAL	69.4	78.6	79.2	68.0	67.8	74.9	13.5	74.2	43.9	63.3
	En-PAL	74.6	78.0	78.8	71.5	69.6	75.0	16.0	74.6	47.3	65.6
	Cross-PAL	82.3	76.9	80.2	75.7	71.6	76.8	37.7	74.5	50.2	69.5
$GPT_{instruct}$	Native-PAL	70.6	79.4	79.0	67.9	69.7	75.4	16.3	75.6	44.0	64.6
	En-PAL	75.3	78.7	79.3	71.8	70.2	75.6	35.0	73.6	45.6	65.9
	Cross-PAL	**82.6**	78.2	**81.6**	**76.8**	**73.1**	**77.2**	40.3	76.1	**53.4**	**70.2**
Llama2-70	Direct	55.9	65.2	64.6	59.8	58.3	68.6	8.5	67.5	37.8	54.0
	Native-CoT	60.7	64.8	60.9	60.5	59.1	67.3	13.2	66.8	36.7	54.8
	En-CoT	63.5	66.3	62.8	61.7	60.2	66.0	20.3	65.9	40.3	56.7
	Cross-CoT	66.0	**69.5**	**65.9**	**64.6**	62.5	**68.6**	**30.7**	**69.3**	42.4	59.8
CodeLlama-70	Native-PAL	61.6	65.0	62.4	60.9	60.7	68.9	16.0	67.9	38.5	55.7
	En-PAL	63.9	67.2	63.7	62.8	61.7	67.6	22.4	66.3	42.0	57.5
	Cross-PAL	**70.6**	69.5	65.8	65.7	**64.3**	66.8	28.0	66.8	**45.9**	60.3
Mixtral8x7	Direct	63.5	**67.5**	64.2	59.7	60.1	68.3	15.1	68.5	38.2	56.2
	Native-CoT	63.1	66.7	65.3	60.2	61.4	**69.5**	15.6	69.3	**40.1**	56.8
	En-CoT	**66.2**	67.3	**66.8**	61.7	62.5	68.9	**16.2**	70.0	40.1	**57.6**
	Cross-CoT	64.8	64.7	65.4	**62.3**	**62.8**	66.2	15.6	**70.3**	37.1	56.5

Table 17: Accuracies (%) on MSVAMP using the reasoning methods described in Appendix C(for each model, we reported best performances per language and per method in **bold**).

J Results Arithmetic Reasoning Tasks Smaller Models

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
						MGSM						
Llama2-7	Direct	48.4	50.2	54.0	56.8	42.0	54.8	28.0	46.2	5.4	38.4	42.5
	Native-CoT	54.8	51.0	55.4	57.6	48.8	58.4	27.4	49.2	20.0	41.6	46.1
	En-CoT	56.0	55.2	56.4	60.2	**51.0**	60.2	30.0	**50.2**	22.6	43.8	48.0
	Cross-CoT	53.8	54.4	**56.2**	**57.6**	50.4	**62.6**	27.4	50.0	28.8	45.2	48.9
CodeLlama-7	Native-PAL	55.0	51.8	56.0	57.8	49.0	59.6	27.8	50.0	22.6	42.8	47.2
	En-PAL	57.0	55.0	56.8	60.4	50.0	61.8	30.6	50.0	24.0	42.0	48.8
	Cross-PAL	**54.2**	**56.0**	55.2	57.0	50.2	62.8	**32.4**	49.8	**29.6**	**45.8**	**49.3**
Mistral-7	Direct	**56.0**	**60.6**	**62.0**	**60.2**	52.0	**60.0**	**34.4**	**52.0**	12.0	47.4	**49.5**
	Native-CoT	54.2	58.4	60.2	58.6	51.4	58.6	32.6	50.2	12.2	47.8	48.0
	En-CoT	55.6	59.2	61.4	59.0	**52.2**	58.8	32.4	51.0	**14.0**	**48.0**	48.4
	Cross-CoT	54.2	57.4	60.0	58.4	50.2	58.6	32.0	51.8	12.4	47.8	48.2
Mistral-7	Native-PAL	53.6	58.0	59.0	58.2	50.6	58.2	33.0	50.0	12.4	47.4	48.0
	En-PAL	55.2	59.4	60.8	59.2	51.2	58.0	32.6	50.2	12.6	46.2	48.4
	Cross-PAL	53.8	57.2	59.0	57.6	49.4	58.0	32.2	52.0	12.6	46.4	47.8
StarCoder2	Direct	50.2	51.8	49.2	50.8	48.0	52.2	16.8	42.6	9.0	41.4	41.6
	Native-PAL	54.6	56.8	52.4	52.6	48.8	54.0	24.6	48.6	14.0	46.8	45.1
	En-PAL	**56.2**	**58.4**	54.0	54.8	**50.2**	56.4	26.2	52.8	16.2	48.0	47.3
	Cross-PAL	54.2	57.2	**54.6**	57.0	50.0	**62.2**	**28.0**	**50.2**	**25.0**	**50.2**	**48.8**
						MSVAMP						
Llama2-7	Direct	51.2	57.3	57.1	51.0	50.9	56.3	10.4	60.2	-	30.1	46.8
	Native-CoT	52.8	58.7	58.2	52.3	51.7	57.0	11.7	62.8	-	32.3	48.6
	En-CoT	55.6	59.8	**60.0**	52.6	**54.2**	56.9	18.8	63.7	-	34.5	51.0
	Cross-CoT	53.4	57.7	58.0	51.6	51.3	57.2	**19.7**	63.4	-	32.3	48.8
CodeLlama-7	Native-PAL	54.0	59.2	58.6	53.0	50.9	56.8	14.5	63.0	-	34.2	49.4
	En-PAL	56.0	**60.4**	59.6	52.8	54.0	57.8	20.0	**64.0**	-	**36.0**	**51.2**
	Cross-PAL	**55.8**	59.4	57.3	**55.4**	54.0	**58.8**	17.5	57.6	-	29.2	49.6
Mistral-7	Direct	**52.6**	58.7	**59.0**	52.3	51.4	55.9	8.8	62.1	-	**32.7**	48.2
	Native-CoT	50.7	57.2	56.8	52.0	52.1	56.8	9.1	**63.7**	-	31.8	47.8
	En-CoT	51.3	**58.6**	57.2	**53.2**	**52.8**	**57.6**	**10.4**	62.1	-	32.3	**48.6**
	Cross-CoT	50.8	57.3	57.6	53.0	52.4	54.3	6.7	59.3	-	28.6	46.2
Mistral-7	Native-PAL	50.2	57.3	56.5	51.5	52.6	55.9	9.4	62.1	-	30.6	47.3
	En-PAL	50.4	57.2	56.5	53.0	51.3	57.0	9.4	60.3	-	30.4	47.2
	Cross-PAL	51.4	58.5	57.9	52.0	52.7	52.4	8.9	60.4	-	29.5	47.0
StarCoder2	Direct	54.4	59.0	57.4	54.2	52.6	58.7	11.6	58.3	-	32.0	48.6
	Native-PAL	56.0	60.0	58.6	**55.8**	52.8	59.2	12.8	58.6	-	32.0	50.0
	En-PAL	56.2	**60.2**	58.2	55.4	**53.2**	**59.0**	14.5	59.2	-	32.7	49.6
	Cross-PAL	**57.0**	59.6	**58.5**	56.3	51.3	57.4	**15.1**	**58.9**	-	**34.2**	**50.2**

Table 18: Accuracies (%) on MGSM and SVAMP of further models using the reasoning methods described in Appendix C (in **bold** the best performance of each model).

K Results Arithmetic Reasoning Tasks GPT-4

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
						MGSM						
GPT-4	Direct	78.0	79.2	83.0	78.4	76.2	82.2	38.8	72.0	18.4	65.4	67.1
	Native-CoT	78.8	79.6	84.2	79.2	77.2	83.4	44.0	76.2	25.4	66.2	68.4
	En-CoT	80.6	80.0	84.4	81.2	78.2	84.2	56.0	78.4	45.6	68.6	73.7
	Cross-CoT	83.0	83.2	85.2	83.4	80.0	83.2	**60.6**	80.6	**57.0**	68.2	76.9
	Native-PAL	79.8	80.2	84.8	79.6	78.2	84.0	41.0	77.2	41.2	66.4	71.2
	En-PAL	80.8	81.4	84.8	80.0	79.2	83.2	55.0	79.2	51.8	69.2	74.3
	Cross-PAL	**84.4**	**83.6**	**85.0**	**83.8**	**81.6**	**85.0**	58.8	**81.2**	56.2	**70.2**	**77.0**
						MSVAMP						
GPT-4	Direct	74.1	73.6	81.2	76.3	70.5	77.2	36.0	70.5	-	65.9	69.2
	Native-CoT	74.6	74.2	81.8	76.2	71.4	78.1	38.0	71.2	-	66.3	70.2
	En-CoT	76.7	76.3	82.6	77.8	71.2	81.3	39.6	71.8	-	67.2	71.6
	Cross-CoT	81.3	77.5	83.4	78.2	73.1	82.1	42.8	73.6	-	68.5	73.4
	Native-PAL	75.8	76.9	83.2	78.0	72.4	79.6	40.2	72.0	-	66.3	71.7
	En-PAL	77.9	78.8	83.2	78.1	72.1	82.4	38.2	72.5	-	69.4	72.5
	Cross-PAL	**82.4**	**78.6**	**83.7**	**78.5**	**73.7**	**82.7**	**43.2**	**74.5**	-	**70.2**	**74.3**

Table 19: Accuracies (%) on MGSM and SVAMP of GPT-4 on first 100 questions for each language using the reasoning methods described in Appendix C.

L Performances on XNLI

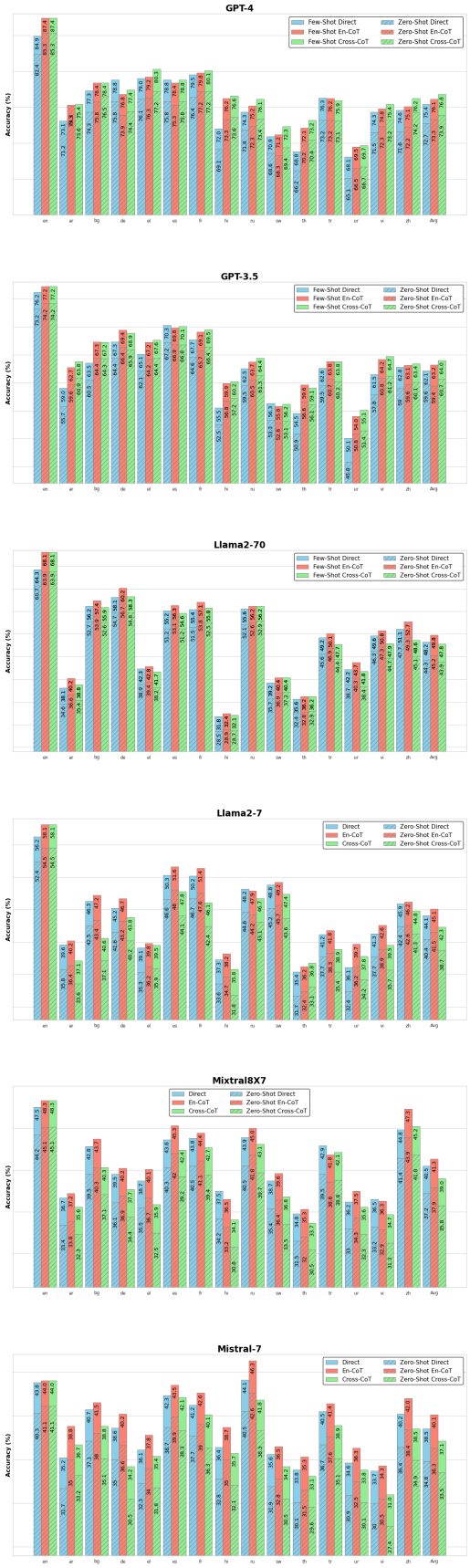

Table 20: Accuracies XNLI using 4-shot and 0-shot prompts.

N Performances on PAWS-X

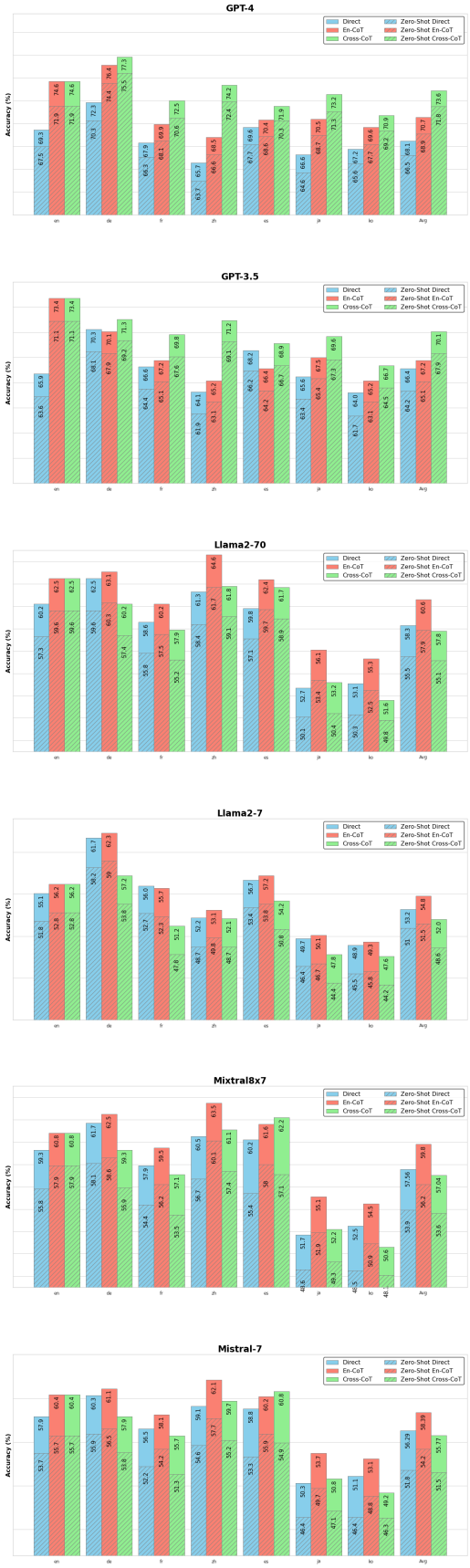

Table 21: Accuracies PAWS-X using 4-shot and 0-shot prompts.

Q Performances on XCOPA

Model	et	ht	id	it	qu	sw	ta	th	tr	vi	zh	Avg
GPT-4												
Direct	**98.8**	93.2	**97.6**	**99.8**	78.6	94.4	79.6	87.8	**97.4**	86.2	92.6	89.0
Native-CoT	98.0	94.6	92.8	98.6	82.0	92.6	82.4	86.0	92.0	84.2	91.6	90.7
En-CoT	95.8	94.0	96.0	98.2	80.0	**95.2**	84.6	88.0	93.4	85.2	93.6	91.7
Cross-CoT	97.8	**95.2**	96.6	95.0	**84.8**	93.8	**85.8**	**91.8**	96.6	**87.2**	**94.0**	**96.2**
GPT-3.5												
Direct	90.6	72.0	90.4	95.2	54.6	82.0	59.0	77.6	91.0	83.6	90.4	80.2
Native-CoT	92.0	79.0	90.4	96.0	81.4	81.8	64.2	81.0	90.2	84.4	93.0	83.8
En-CoT	92.4	78.2	91.6	**96.8**	81.4	81.6	**64.8**	80.2	93.6	85.2	**94.0**	85.3
Cross-CoT	**94.0**	**79.6**	**92.2**	96.4	**82.6**	**82.0**	63.2	**82.0**	**93.8**	**86.0**	93.4	**84.4**
Mixtral8x7												
Direct	**82.5**	**68.0**	**81.6**	54.5	**83.1**	60.3	78.1	81.9	**80.5**	74.2	**70.6**	**74.2**
Native-CoT	80.8	65.2	78.5	55.0	80.1	60.2	80.0	81.0	78.4	70.2	70.6	72.7
En-CoT	81.7	66.5	79.3	53.5	82.9	61.3	**80.8**	**82.4**	79.8	**74.7**	70.3	73.9
Cross-CoT	80.7	67.1	77.3	**54.2**	82.0	**60.7**	80.2	80.3	79.2	73.5	69.2	73.1
Llama2-70												
Direct	80.4	66.2	79.8	82.4	52.8	81.6	58.4	76.0	79.2	73.0	69.2	70.2
Native-CoT	83.0	68.0	**81.2**	83.4	55.0	82.3	60.2	77.8	**81.0**	**76.2**	**72.4**	73.5
En-CoT	**84.2**	**68.8**	80.4	**84.6**	**55.2**	**82.8**	**60.6**	**78.4**	80.4	74.4	71.6	**75.1**
Cross-CoT	79.8	66.0	78.2	81.6	51.2	80.2	57.8	77.2	80.4	73.6	70.6	72.7
Llama2-7												
Direct	39.6	32.5	58.4	55.8	47.2	34.6	47.4	33.2	43.0	59.6	50.4	45.4
Native-CoT	**42.0**	**37.2**	**62.4**	**58.0**	48.0	**37.0**	48.0	33.0	44.0	60.2	50.2	46.0
En-CoT	42.8	36.6	60.2	56.2	**50.0**	36.8	**48.6**	**34.8**	**44.2**	**60.8**	**51.6**	**47.1**
Cross-CoT	40.8	36.2	57.8	56.2	48.4	33.0	47.0	34.4	44.2	60.2	51.6	46.2
Mistral-7												
Direct	**42.6**	36.5	60.1	57.8	48.7	37.3	49.2	36.6	45.2	59.3	51.2	47.7
Native-CoT	42.4	**37.6**	58.2	**58.6**	**52.0**	37.8	49.6	37.4	46.0	**60.4**	**54.0**	46.6
En-CoT	41.9	37.1	**59.8**	57.2	50.1	38.2	**49.7**	**38.5**	**46.3**	60.1	52.3	**48.2**
Cross-CoT	39.7	36.5	57.6	56.8	49.6	**38.4**	48.7	37.5	45.2	59.4	50.7	47.3
HUMAN (Ponti et al., 2020)	98.2	96.4	100.0	97.0	94.8	99.0	98.6	98.2	96.4	98.4	96.6	97.6

Table 22: Accuracies (%) on XCOPA (Ponti et al., 2020) using the reasoning methods described in Appendix C. (Direct, Native-CoT, En-CoT and Cross-CoT as introduced in Section 3.3).

R Performances on English

Model	Method	MGSM	MSVAMP	XNLI	PAWS-X	XCOPA
GPT-4	Direct	94.6	92.5	84.9	69.3	98.6
	CoT	96.8	95.3	87.4	74.6	**99.4**
	PAL	**97.2**	**96.7**	-	-	-
GPT-3.5	**Direct**	80.6	82.7	77.2	65.9	94.5
	CoT	84.8	85.2	**76.6**	**73.4**	**95.0**
	PAL	**86.6**	**86.3**	-	-	-
Llama-70	Direct	70.2	73.7	64.3	60.2	85.6
	CoT	71.8	75.3	**68.1**	**62.5**	**85.9**
	PAL	**72.4**	**76.9**	-	-	-
Llama-7	Direct	64.6	68.5	56.2	55.1	**60.8**
	CoT	67.8	69.4	**58.1**	**56.2**	60.6
	PAL	**69.2**	**70.1**	-	-	-
Mixtral8x7	Direct	76.0	78.0	47.5	59.3	66.2
	CoT	75.4	77.2	**48.3**	**60.8**	**67.1**
	PAL	77.2	77.8	-	-	-
Mistral-7	Direct	66.2	67.8	43.8	57.9	**62.4**
	CoT	66.8	66.9	**44.0**	**60.4**	61.6
	PAL	**67.2**	**67.5**	-	-	-
Starcoder2	Direct	58.0	61.4	-	-	-
	PAL	**64.2**	**63.9**	-	-	-

Table 23: Evaluations on proposed tasks using CoT and PAL of English versions of proposed task.

S Qualitative Analysis

Language Generated The *reasoning methods* introduced in Section 3.3 elicit Large Language Models to generate answers following in-context demonstrations. Specifically, operating in multilingual scenarios, in-context demonstrations were provided in several different languages; hence, the models are expected to be able to generate language-specific responses following the examples provided in context.
To analyze the language compositions in answers delivered from the different models, we use the OpenLID[4] library (Burchell et al., 2023). In particular, we focus the analysis on CoT generations (particularly on `Native-CoT` and `En-CoT`). In both cases, we apply sentence splitting. We then apply OpenLID to the responses generated downstream of the CoT method by attributing the most frequent language present to each response. In contrast, for PAL, we start with sentence splitting but remove possible code fragments (found by string matching between the sentence and a list consisting of symbols such as '=', '*', '+', '-') from the analysis. Following this superficial cleaning, we analyze the composition of the response by attributing the maximum language present in the generated sentences.

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
GPT-4	Native-CoT	88.2 (75)	86.5 (76)	88.2 (82)	80.6 (77)	54.7 (46)	84.3 (78)	46.0 (39)	69.3 (70)	55.5 (35)	57.2 (38)	73.9 (58.9)
	En-CoT	44.2 (45)	36.3 (40)	48.7 (48)	25.4 (36)	18.2 (22)	48.1 (51)	28.8 (30)	36.3 (29)	23.1 (32)	36.4 (27)	36.2 (39)
GPT-3.5	Native-CoT	96.3 (78)	92.6 (78)	94.5 (85)	78 (64)	55 (26)	80 (82)	44 (32)	67 (18)	53 (24)	60 (28)	72.0 (48.8)
	En-CoT	40.2 (70)	30.7 (75)	42.3 (76)	20.4 (67)	16.8 (58)	46.8 (73)	23.2 (56)	31.4 (49)	19.6 (47)	53.5 (51)	34.6 (65)
Llama-70	Native-CoT	72.8 (66)	76.5 (64)	80.2 (69)	63.6 (62)	55.2 (46)	63.6 (43)	36.7 (33)	38.2 (37)	46.5 (35)	38.2 (29)	66.5(54)
	En-CoT	18.8 (40)	16.7 (45)	33.5 (41)	19.8 (37)	13.2 (26)	36.3 (33)	17.9 (17)	19.5 (16)	11.8 (21)	37.9 (19)	24.3(44)
Llama-7	Native-CoT	67.6 (55)	62.8 (63)	60.5 (68)	58.3 (52)	46.2 (18)	50.3 (27)	30.0 (26)	26.8 (35)	40.2 (28)	32.8 (16)	41.4(55)
	En-CoT	16.7 (32)	17.2 (40)	32.6 (37)	17.5 (35)	16.6 (23)	33.6 (39)	14.9 (44)	17.5 (38)	27.8 (50)	30.9 (36)	22.6(38)

Table 24: Percentage (%) of answers generated in `Native` language (i.e., the specific language of the examined sub-task) for MGSM using the reasoning methods described in Appendix C. Moreover, we reported the accuracies (values in brackets) for each set of generations in `Native` language.

Number of Hops Analyzing the composition of languages in the answers provided by the different models is useful to understand whether a certain model follows the in-context prompts by generating language-specific answers and, if so, what the error rate is. However, it is important to analyze the composition of the provided answers. To qualitatively estimate the generated responses, we propose the analysis of the phrases present in the responses generated by the models under study. In particular, given an answer A, composed of a set of sentences $(\{s_1, s_2, \ldots, s_n\})$, we define $Hops$ as the number of sentences the models generate to deliver the solution. Since the in-context rationales provided have an average number of 4 $Hops$ (min value 3 and max value 5) (Shi et al., 2022), they do not include the final keyword *"Answer:"* or *"The answer is:"*, we do not consider the final keyword for a more realistic value as it often repeats the last sentence. Formally, let A be composed of n sentences and represent the final answer. The sum of sentences in A gives the total number of $Hops$.
Hence, we compute this value for the generations of models analyzed and report results in the following table.

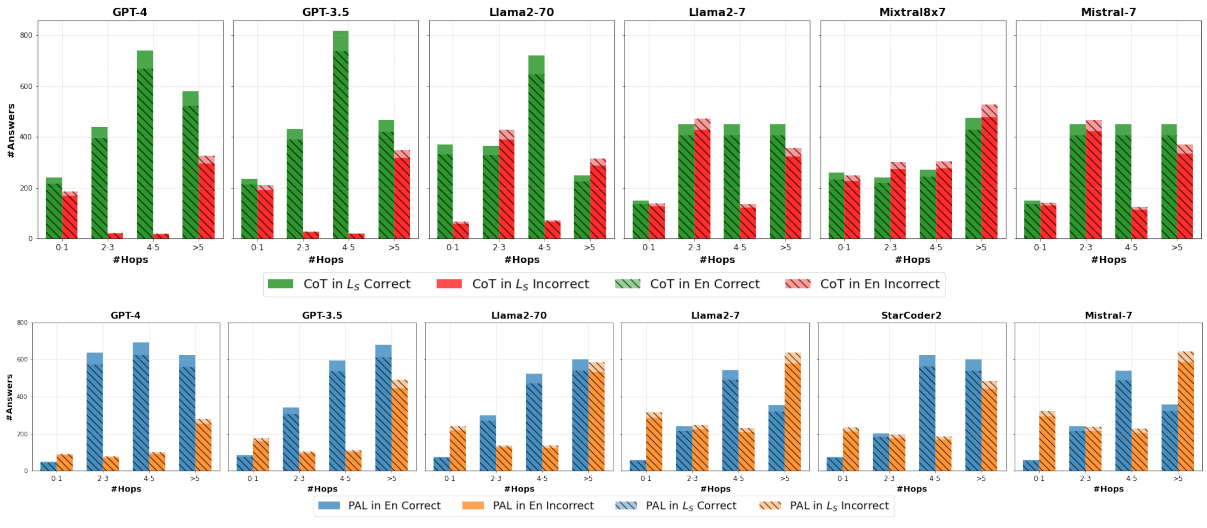

Table 25: Number of $Hops$ generated for each model introduced in Section 3.2

T Scalability to further LLMs

We study the performance of additional Large Language Models (LLMs) that are not considered in the main analysis. The models are chosen for performances in mathematical reasoning tasks (in the case of WizardMath (Luo et al., 2023)) or in specific languages beyond English (Tulu (Ivison et al., 2023) and Yi (AI et al., 2024)), and finally for abilities in functions with a limited number of parameters (gemma (Team et al., 2024)). We used the same experimental setup of Section 3. We produced evaluations for a few instances of the MSGM task (we used the same instances as those used for GPT-4). This experiment observes whether the selected models perform comparably to those discussed in Section 4. Figure 26 confirms the results obtained from previous LLMs (results detailed in Table 18), and the following points emerge: **(i)** Reasoning methods operate beyond English. As discussed in Section4.1, LLMs prompted via `En-CoT` stably overperform the baselines, i.e., `Direct`. **(ii)** Ehere are limitations, as yet discussed in Section 4.2. Models with fewer parameters (see Orca and Gemma) underperform when the quality of in-context prompts is more articulated (`Direct` vs. `CoT` case). **(iii)** While the smaller models appear not to benefit under varying in-context demonstrations, the larger models (in these experiments, they are average LLMs not comparable to GPT-4) outperform when the `Cross-CoT` prompting strategy is used, as happens to the results discussed in Section 4.

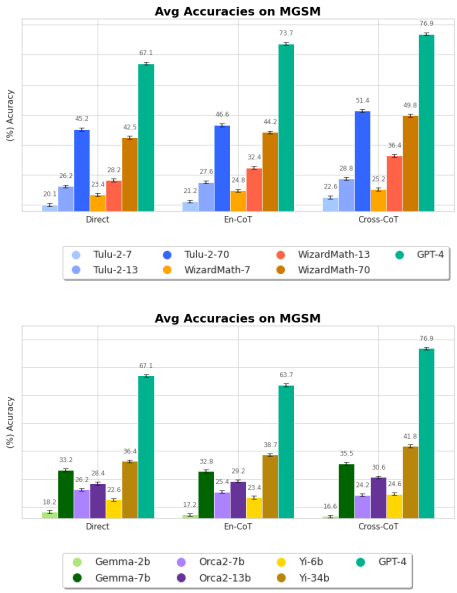

Table 26: Average accuracies across specific languages of further LLMs by using prompting pipelines proposed in Section 3.3.

U Models Vesions

Model	Version
Llama-2-7	meta-llama/Llama-2-7b
Llama-2-13	meta-llama/Llama-2-13b
Llama-2-70	meta-llama/Llama-2-70b
gemma-2	google/gemma-2b
gemma-7	google/gemma-7b
Orca-2-7	microsoft/Orca-2-7b
Orca-2-13	microsoft/Orca-2-13b
Mistral-7-instruct	mistralai/Mistral-7B-Instruct-v0.2
Mixtral	TheBloke/Mixtral-8x7B-Instruct-v0.1-GPTQ
Yi-6b	TheBloke/Yi-6B-GPTQ
Yi-34b	TheBloke/Yi-6B-GPTQ
Tulu-2-7	TheBloke/tulu-2-7B-GPTQ
Tulu-2-13	TheBloke/tulu-2-13B-GPTQ
Tulu-2-70	TheBloke/tulu-2-70B-GPTQ
WizardMath-7	TheBloke/WizardMath-7B-V1.0-GPTQ
WizardMath-13	TheBloke/WizardMath-13B-V1.0-GPTQ
WizardMath-70	TheBloke/WizardMath-70B-V1.0-GPTQ
StarCoder2	bigcode/starcoder2-15b
CodeLlama-70 (7)	TheBloke/CodeLlama-70B (7)-Instruct-GPTQ
GPT-3.5-turbo	OpenAI API (gpt-3.5-turbo-0125)
GPT-instruct	OpenAI API (gpt-3.5-turbo-instruct)
GPT-4	OpenAI API (gpt-4-1106-preview)

Table 27: List the versions of the models proposed in this work, which can be found on huggingface.co. We used the configurations described in Appendix F in the repositories for each model *(access to the following models was verified on 14 June 2024).

Figure 2 complete

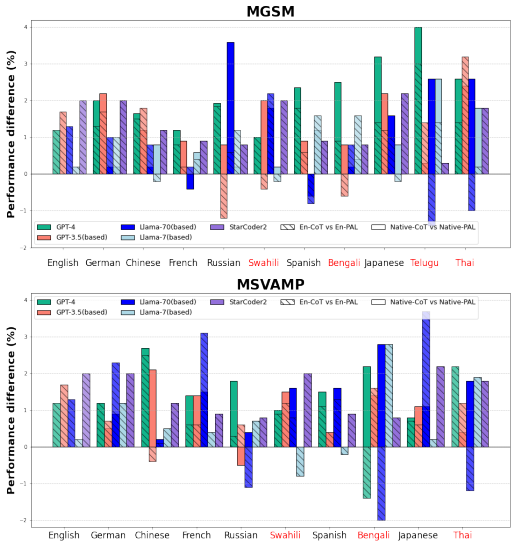

Table 28: Performance difference between `Native-PAL` and `Native-CoT` for each individual language (low-resource languages in red) in MGSM and MSVAMP (we also reported the difference between `En-PAL` and `En-CoT` hatched). This is Figure 2 extended for each model.

V Related Works

work	methods	models	tasks	findings
(Shi et al., 2022)	Direct, CoT, Native-CoT	GPT-3, PaLM	MGSM	multilingual- reasoning
(Huang et al., 2023)	En-CoT, Translate-CoT, Meta-prompting∗	GPT-3.5 Llama2-70	MGSM, XCOPA, MKQA PAWS-X, XNLI, FLORES	in-context translation En-CoT matter
(Qin et al., 2023)	Cross-CoT∗, En-CoT, Native-CoT	GPT-3.5 PaLM	MGSM, XCOPA PAWS-X, XNLI	English lead Cross-lingual alignment
(Ranaldi et al., 2024)	Cross-ToT	GPT-3.5 Llama2-s	MGSM, XCOPA PAWS-X, XNLI	ToT lead self-consistent reasoning
OUR	En-,Native-, Cross-, CoT and PAL	GPT-s, Llama2-s Mistral-s, StarCoder	MGSM, SVAMP PAWS-X, XNLI, XCOPA	

Table 29: Summary related works that propose multilingual reasoning in-context approaches (i.e., no fine-tuning), ∗ means double-step prompting phase, where not specified there are single-step prompting approach.

Native-CoT

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。
Q: 问题：服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑?
A: 从周一到周四有4 天。每天增加5 台电脑。这意味着一共增加了4 * 5 = 20 台电脑。一开始有9 台电脑，所以现在有9 + 20 = 29 台电脑。答案是29。
Q: 问题：肖恩有五个玩具。圣诞节他从他爸爸妈妈那里各得到了两个玩具。他现在有多少个玩具?
A: 他有5 个玩具。他从妈妈那里得到2 个，所以之后他有5 + 2 = 7 个玩具。然后他又从爸爸那里得到了2 个，所以他一共有7 + 2 = 9 个玩具。答案是9。
Q: 制作一件袍子需要2 匹蓝色纤维布料和这个数量一半的白色纤维布料。它一共需要用掉多少匹布料?
让我们逐步思考！为清晰起见，答案应采用以下格式："答案：[数字]"。

GPT-4

制作一件袍子需要2匹蓝色纤维布料和这个数量一半的白色纤维布料。首先，2匹的一半是1匹，所以需要1匹白色纤维布料。将蓝色和白色纤维布料相加，2匹蓝色加上1匹白色总共是3匹布料。
答案：3

GPT-3.5

首先，袍子需要2匹蓝色纤维布料。然后，它需要同样数量的白色纤维布料的一半，也就是1匹白色纤维布料。所以总共需要用掉2 + 1 = 3匹布料。
答案：3

Mixtral8x7

To make one robe, you need 2 pieces of blue fabric and half as much white fabric. Half of the amount of white fabric would be 2/2 = 1 piece. Therefore, to make one robe, you will need a total of 2 (blue) + 1 (white) = 3 pieces of fabric. Answer: 3

Llama2-7

[...] 制作一件袍子需要 two π blue cotton fabrics and this amount half white cotton fabric. How much fabric do they need in total? A: To make one shirt, you need 2 pi (π) blue cotton fabrics and half of them is equal to 1 pi (π) white cotton fabric. So, in total, you need 2 * π + (1/2) * π = 3π white cotton fabrics.

Mistral-7

To make one suit, it requires 2 pieces of blue fabric and half as many pieces of white fabric. Therefore, it takes 2 (blue) + 2 (half of what 2 is for white) = 3.5. Since we cannot have a half piece of fabric, let's assume it needs an additional 0.5 pieces to cover this requirement. So, it takes a total of 3.5 x 2 = 7 pieces of fabric to create one suit. The answer is 7.

Table 30: Example output answers generated by different LLMs prompted via `En-CoT` (Section 3.3). In general, LLMs produce good answers while Smaller LLMs (for example see Llama2-7) produce misleading answers.

.

Cross-CoT

Given the following examples, please act as an expert in multilingual understanding in Chinese.
Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。
Q: 问题：服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑?
A: 从周一到周四有4 天。每天增加5 台电脑。这意味着一共增加了4 * 5 = 20 台电脑。一开始有9 台电脑，所以现在有9 + 20 = 29 台电脑。答案是29。
Q: 问题：肖恩有五个玩具。圣诞节他从他爸爸妈妈那里各得到了两个玩具。他现在有多少个玩具?
A: 他有5 个玩具。他从妈妈那里得到2 个，所以之后他有5 + 2 = 7 个玩具。然后他又从爸爸那里得到了2 个，所以他一共有7 + 2 = 9 个玩具。答案是9。
Q: 问题：利亚有32 块巧克力，她妹妹有42 块。如果她们吃了35 块，她们一共还剩下多少块?
.....
Q: 问题：如果停车场里有3 辆车，又来了2 辆车，停车场里有多少辆车?
.....
Q: 制作一件袍子需要2 匹蓝色纤维布料和这个数量一半的白色纤维布料。它一共需要用掉多少匹布料?
After understanding, act as an expert in arithmetic reasoning in English. Let's answer the question step-by-step! For clarity, the answer should have the following format:'Answer:[num]'

GPT-4

To solve the question regarding the amount of fabric needed for the robe, [...] Here's the step-by-step calculation:
Total fabric = Blue fabric + White fabric = 2 bolts (blue) + 1 bolt (white) = 3 bolts
Thus, the total amount of fabric needed to make one robe is 3 bolts.
Answer: 3

GPT-3.5

To solve the problem, we need to determine the total number of pieces of fabric required to make a robe.
Given that making one robe requires 2 pieces of blue fabric and half that amount of white fabric, we can calculate:
Blue fabric needed: 2 pieces.
White fabric needed: Half of 2 pieces, which is 1 piece.
Adding the blue and white fabric together, we find that the robe requires 2 + 1 = 3 pieces of fabric in total.
Answer: 3

Mixtral8x7

A robe requires 2 pieces of blue fabric and half the amount of white fabric as compared to the blue fabric.
First, determine the amount of white fabric required. Since the white fabric is half the amount of blue fabric, and there are 2 pieces of blue fabric, the robe needs 0.5 * 2 = 1 piece of white fabric.
Next, add up the amount of both types of fabric to find the total amount required. The robe needs 2 pieces of blue fabric and 1 piece of white fabric, so it needs 2 + 1 = 3 pieces of fabric in total.
Therefore, a robe requires 3 pieces of fabric in total (2 pieces of blue fabric and 1 piece of white fabric).

Llama2-7

Step 1: First, let's calculate the amount of blue fabric needed:
2 yards x 2 = 4 yards
Step 2: [...]
8 + 4 = 12 yards Answer: 12 yards

Mistral-7

To make one suit, you need 2 pieces of blue fabric and half as many pieces of white fabric. So you need 2 (blue) + 2 (half of the number of white pieces) = 2 + 1 = 3 pieces of blue fabric and 2 (half of the number of white pieces) = 2 pieces of white fabric in total. Therefore, you need 3 (blue) + 2 (white) = 5 pieces of fabric in total. The answer is 5.

Table 31: Example output answers generated by different LLMs proposed in our work prompted via Cross-CoT (Section 3.3). In contrast to the generation in Table 30 Llama-2-7 does not deliver misleading answer.

.

Cross-PAL

Given the following examples, please act as an expert in multilingual understanding in German.
Q: Jason hatte 20 Lutscher. Er hat Denny einige Lutscher gegeben. Jetzt hat Jason 12 Lutscher. Wie viele Lutscher hat Jason Denny gegeben?
A:

```
#solution in Python
    # Jason hatte 20 Lutscher.
    jason_lollipops_initial = 20
    # Jetzt hat Jason 12 Lutscher.
    jason_lollipops_left = 12
    # Lutscher werden Denny
    lollipops_given_to_denny = jason_lollipops_initial - jason_lollipops_left
    # Die Antwort ist
    return lollipops_given_to_denny
```

Die Antwort ist 11.
(...more in-context demonstrations...)
Q: Kylar geht ins Kaufhaus, um Gläser für seine neue Wohnung zu erwerben. Ein Glas kostet 5 US-Dollar, aber jedes weitere Glas kostet nur 60% des Ausgangspreises. Kylar möchte 16 Gläser kaufen. Wie viel muss er dafür ausgeben?
A: After understanding, you should act as a programmer in English and answer the question. For clarity, the answer should have the following format: 'The answer is [num]'.

GPT-4

A: #solution in Python

```
def total_cost_for_glasses():
    first_glass_price = 5
    subsequent_glass_price = 0.60 * first_glass_price
    total_glasses = 16
    cost = first_glass_price + (total_glasses - 1) * subsequent_glass_price
    return cost
```

The answer is 16

GPT-3.5

To calculate the total cost for Kylar's purchase of 16 glasses, we can break it down into two parts: the cost of the first glass and the cost of the additional glasses.
The cost of the first glass is $5.
For each additional glass beyond the first one, the cost is 60% of the original price, which is 0.6 * $5 = $3.
So, for the remaining 15 glasses, the cost would be 15 * $3 = $45.
Adding the cost of the first glass to the cost of the additional glasses, the total cost would be $5 + $45 = $50.
Therefore, Kylar would need to spend $50 to purchase 16 glasses for his new apartment.

CodeLlama-70

A: #solution in Python

```
def cost_of_glasses(glasses):
"""A function to calculate the cost of glases."""
    glasses_cost = 0
    glass_price = 5
    for glass_index in range(glasses):
      glasses_cost += glass_price
      glass_price = 0.6 * glass_price
    return glasses_cost
```

The answer is 16

CodeLlama-7

A: #solution in Python

```
def glass_cost(n): return 5 * n + 0.6 * 5 * (n - 1) price = glass_cost(16) print("Kylar will spend around $154.00 to buy 16 glasses.")
```

CodeLlama-7

A: #solution in Python

```
def cost_of_glasses(n):
    if n == 1:
       return 5 *.6
    else:
       print("The total cost of", end=" ")
    n = int(input("how many glasses do you want to buy?"))
    total_cost = cost_of_glasses(n)
    print("the purchase will cost:", total_cost)
```

The answer is 16

Table 32: Example output answers generated by different LLMs prompted via Cross-PAL (Section 3.3). Generations are structured and guided by program languages.

W Detailed Structures of Cross-based Methods

Zero- & Few-shot Chain-of-Thought

Q: <Question$_1$> (in En)
A: <CoT demonstration$_1$> (in En)
.....
Q: <Question$_n$> (in En)
A: <CoT demonstration$_n$> (in En)

Q: <Question> (in En)
A: Let's think step-by-step!

English Chain-of-Thought (En-CoT)

Q: <Question$_1$> (not in En)
A: <CoT demonstration$_1$> (in En)
.....
Q: <Question$_n$> (not in En)
A: <CoT demonstration$_n$> (in En)

Q: <Question> (not in En)
A: Let's think step-by-step!

Cross **Chain-of-Thought**

Given the following examples, please act as an expert in multilingual understanding in Chinese.

....*few-examples*.... (in case of few-shot prompting)

Q: 罗杰有5 个网球。他又买了2 罐网球。每罐有3 个网球。他现在有多少个网球?
A: 罗杰一开始有5 个球。2 罐各3 个网球就是6 个网球。5 + 6 = 11。答案是11。

Q: 服务器机房里有九台电脑。从周一到周四，每天又安装了五台电脑。服务器机房里现在有多少台电脑?
A: After understanding, act as an expert in arithmetic reasoning in English. Let's answer the question step-by-step!

Table 33: The tables on the left represent the standard Chain-of-Thought (En-CoT) prompting in a few-shot (Wei et al., 2023) or zero-shot (Kojima et al., 2023) settings. Then, following Shi et al. (2022), we propose En-CoT by using Q in the specific language and the traditional CoT. The table on the right represents Cross-lingual prompting (for easier understanding, we specified the target language, Chinese). Unlike previous settings, the (Q,A) pairs are in the same specific language, but the final answer should be in English.

Program-Aided Language Models (PAL)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 tennis balls.
tennis_balls = 5
2 cans of 3 tennis balls each is
bought_balls = 2 * 3 tennis balls.
The answer is
answer = tennis_balls + bought_balls
The answer is 11

Q: Jason had 20 lollipops. He gave Denny some lollipops. Now Jason has 12 lollipops. How many lollipops did Jason give to Denny?
A:

Cross **Program-Aided Language Models**

Given the following examples, please act as an expert in multilingual understanding in German.

Q: Roger hat 5 Tennisbälle. Er kauft noch 2 Dosen Tennisbälle. In jeder Dose sind 3 Tennisbälle. Wie viele Tennisbälle hat er jetzt?
A: # Roger begann mit 5 Tennisbällen.
tennis_balls = 5
# 2 Dosen mit je 3 Tennisbällen sind
bought_balls = 2 * 3 tennis balls.
# Die Antwort ist
answer = tennis_balls + bought_balls
# Die Antwort ist 11

Q: Gretchen hat 110 Münzen. Es sind 30 mehr Gold- also Silbermünzen. Wie viele Goldmünzen hat Gretchen?
A: After understanding answer the question, you should act as a programmer in English.

Table 34: The table on the left represents Program-Aided Language Models (PAL) method in a few-shot setting where demonstrations of PAL answers are provided as input (Gao et al., 2022; Cheng et al., 2023). On the right, the Cross-lingual PAL (Cross-PAL) prompting where the question and relative answers are in a specific language as proposed for Cross-CoT in Table 33.