Empowering Multi-step Reasoning across Languages via Program-Aided Language Models

Leonardo Ranaldi [†] **Giulia Pucci** [‡] **Barry Haddow** [†] **Alexandra Birch**[†]
(†) School of Informatics, University of Edinburgh, UK
(‡) Department of Computing Science, University of Aberdeen, UK
{first_name.last_name}@ed.ac.uk

Abstract

In-context learning methods are commonly employed as inference strategies, where Large Language Models (LLMs) are elicited to solve a task by leveraging provided demonstrations without requiring parameter updates. Among these approaches are the *reasoning methods*, exemplified by Chain-of-Thought (CoT) and Program-Aided Language Models (PAL), which encourage LLMs to generate reasoning steps, leading to improved accuracy. Despite their success, the ability to deliver multi-step reasoning remains limited to a single language, making it challenging to generalize to other languages and hindering global development.

In this work, we propose Cross-lingual Program-Aided Language Models (`Cross-PAL`), a method for aligning reasoning programs across languages. Our method delivers programs as intermediate reasoning steps in different languages through a double-step cross-lingual prompting mechanism inspired by the Program-Aided approach. Moreover, we introduce Self-consistent `Cross-PAL` (*S*`Cross-PAL`) to ensemble different reasoning paths across languages. Our experimental evaluations show that `Cross-PAL` outperforms existing methods, reducing the number of interactions and achieving state-of-the-art performance.

1 Introduction

Reasoning methods, best exemplified by Chain-of-Thought (CoT) (Kojima et al., 2023), Program-Aided Language Models (PAL) (Gao et al., 2022) (or program-of-thoughts (Chen et al., 2023b)) elicit Large Language Models (LLMs) to generate reasoning paths, thus increasing accuracy. The success of these methods is due to the LLMs' abilities to perform tasks by following in-context structured patterns, a technique known as in-context learning (Brown et al., 2020).

Since using *reasoning methods* brings clear benefits to the final performance, a series of works began to investigate whether these findings could also be transferred to languages other than English. Shi et al. (2022) introduced the first multilingual benchmark to assess the mathematical reasoning abilities of LLMs in different languages, which Huang et al. (2023) exploited to demonstrate the in-context cross-lingual capabilities of larger LLMs. In parallel, Fang et al. (2022) introduced a pre-translation phase to align and improve generative reasoning beyond English. Qin et al. (2023) proposed task-specific solver prompting, using a succession of prompts, eliciting the models to understand questions and deliver CoT answers in specific languages. Finally, Ranaldi et al. (2024d) proposed a single-step pipeline, getting closer to the zero-shot principle but observing the practical functionalities only in models with a few billion parameters.

However, limited attention has been given to the study of the natural language structure of in-context demonstrations and their possible effect on LLMs' multilingual reasoning. Additionally, the results achieved by previous works are based on the capabilities of larger-scale LLMs, which often have restricted availability and inconvenient access costs. This discourages using models with fewer parameters but simultaneously limits the potential benefits users could gain by maximizing the operability of in-context learning.

In this paper, we propose Cross-lingual Program-Aided Language Models (`Cross-PAL`), a method for aligning reasoning programs across languages. `Cross-PAL`, by exploiting the structure of in-context demonstrations, delivers program-of-thought as intermediate reasoning passages in different languages by employing a cross-lingual prompting mechanism inspired by the Program-Aided approach. In addition, we introduce Self-consistent `Cross-PAL` (*S*`Cross-PAL`), which is inspired by the Self-consistent CoT (Wang et al.,

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 12171–12187
November 12-16, 2024 ©2024 Association for Computational Linguistics

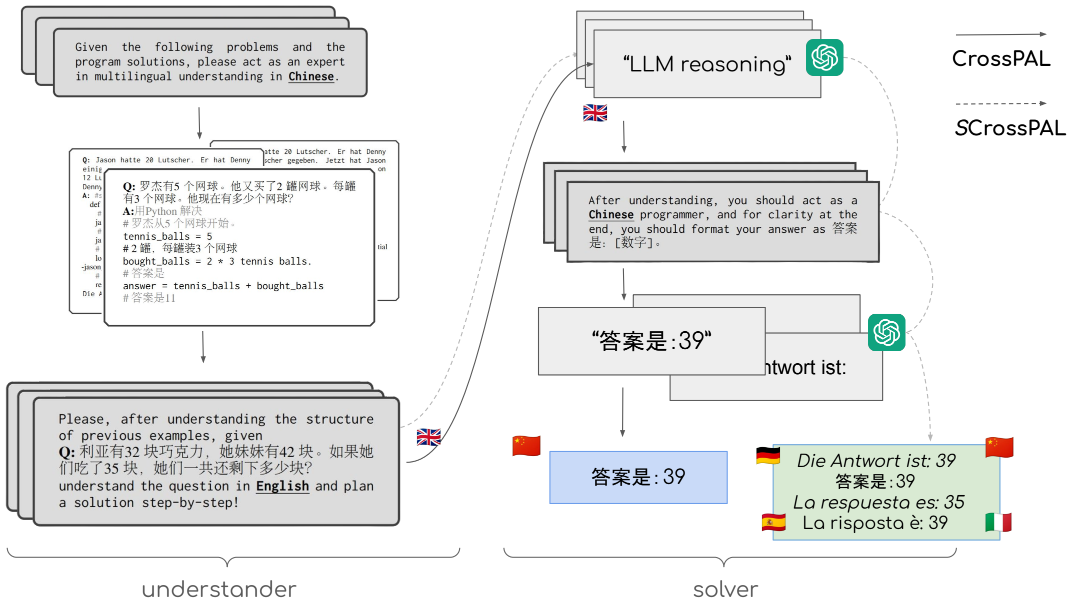

Figure 1: Cross-PAL elicits the LLM to generate reasoning programs across different languages (§2.2). In this example, given separate problems in language L_S (*Chinese*), the conducted steps for solving it are: (1) generate a structured planning strategy in *English* (using in-context demonstrations in L_S), (2) collect the planned strategy and finalize the solution in L_S (that is the language of the original problem). Self-consistent Cross-PAL (*S*Cross-PAL) aligns different reasoning paths, ensembling the final solution (§2.3).

2023), to ensemble different reasoning paths across languages and align them in a unique final solution. We then show that our method works even on small-scale models through a series of analyses.

The key points that lead our contribution are:

- Understanding to what extent in-context structured demonstrations influence the planning abilities in mathematical reasoning tasks.
- Which are the effects emerging during the generation process between different languages, particularly in the self-consistent version of Cross-PAL.
- The operability and scalability of proposed methods on smaller-scale LLMs.

To analyse these points, we operate via Cross-PAL, a novel cross-lingual prompting strategy to bridge the gap across different languages. In particular, we elicit the model to deliver reasoning programs operating between languages using the prompting mechanisms shown in Figure 1. Moreover, we ensemble the answers along each specific language using *S*Cross-PAL, enabling the model to have different views of reasoning paths across languages. We evaluate the impact of our methods on GPT-based models (OpenAI, 2023) using Multilingual Grade School Math (MGSM) (Shi et al., 2022) to have a comparative result with the previous works and MSVAMP (Chen et al., 2023a). Moreover, to analyse the potential scalability of smaller LLMs, we introduce further models best exemplified by Phi-3 (Abdin et al., 2024), Llama3-8 (Touvron et al., 2023), and finally Llama2-7 to have a term of comparison with the previous analyses (Qin et al., 2023).

Experimental results reveal that Cross-PAL outperforms the baselines and achieves robust performances in different languages. Furthermore, the proposed method is scalable and effective even on small-scale models. The main contributions of this work are concluded as follows:

- We introduce Cross-PAL, a novel cross-lingual prompting mechanism that elicits the model to structure solutions to mathematical problems using in-context reasoning programs. We show that structured demonstrations facilitate the planning of reasoned solutions and increase final performances.
- We further propose *S*Cross-PAL, which is self-consistent prompting that allows the integration of reasoning paths across different languages. This mechanism enables the alignment of multilingual abilities by strengthening deficient pathways.

- Finally, we provide extensive evaluations of the role of each language by comparing the effects of low-resource and high-resource languages. Hence, we demonstrate that our approaches can effectively improve multilingual performance in mathematical reasoning tasks and, in contrast to the previous works, achieve stable, robust results in different scales of LLMs.

2 Multi-step Reasoning across Languages

To elicit the reasoning abilities of Large Language Models (LLMs) across languages, we propose Cross-PAL, which is a cross-lingual alignment reasoning program, as a solution. In particular, our method goes beyond the previous multilingual prompting strategies introduced in §2.1. Our approach, which takes inspiration from (Gao et al., 2022), uses reasoning programs, also defined as programs-of-thought (Chen et al., 2023b), to structure the generation and elicit the LLMs to deliver structured answers as discussed in § 2.2.

2.1 Cross-lingual Chain-of-Thoughts

Cross-lingual alignment is a strategic challenge for enabling cross-lingual transfer. Shi et al. (2022) proposed a series of prompts to elicit models to generate CoT answers in specific language `Native-CoT`, and in English `En-CoT` and `Translate-CoT` (detailed in Table 10). Later, Huang et al. (2023) produced a single prompt template that showed the in-context cross-lingual sensitivity of the more robust LLMs. Qin et al. (2023) extends the previous works method on two phases: Cross-lingual alignment prompt and task-specific solver prompting by using two separate steps, as shown in Table 11, to handle input and output in different languages.

Although this second approach overcomes the limitations of previous works that emerged in low-resource languages, it is based on a double-step prompt that requires hand-crafted demonstrations that do not benefit the downstream performances of smaller-scale LLMs.

Hence, Ranaldi et al. (2024d) proposed Cross-ToT, which aims to centralize cross-lingual reasoning in a single pathway by generating self-consistent CoT as reported in Table 12.

2.2 Cross-lingual Program-Aided Reasoning

To elicit the cross-lingual reasoning ability of LLMs, we operate the structure of the in-context prompt by transferring the PAL (Gao et al., 2022; Chen et al., 2023b) methodology in cross-lingual scenarios (Cross-PAL). Our approach consists of a double-step prompt structured in two steps: first, there is a (i) cross-lingual code-switching understanding phase (named *"understander"*) followed by a (ii) language-specific resolution phase (named *"solver"*).

Cross-lingual Understander To elicit LLMs to understand the provided problem and plan a solution by using in-context structured demonstrations in a specific language "[Language L_s]", we plan the prompt in the following way:

```
Given the following problems and the program solutions, please act as an expert in multilingual understanding in [Language Ls].
```

To establish the LLMs' expertise in multilingual comprehension, we introduce program-of-thought demonstrations consisting of code-like intermediate steps to elicit LLMs' handling of the question and structuring of the solution. In the main setting, we use two demonstrations that are composed of a question in [Language L_s] and a program-like solution in pseudo-code commented in a specific language (here we report a single demonstration; see Appendices F and K for additional examples):

```
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 11
```

We conclude the first phase by eliciting a step-by-step understanding process to align the planning paths from the original language [Language L_s] to the target language [Language L_t].

```
Please, after understanding the structure of previous examples, given
Q: [question Q (in Ls)]
understand the question in [Language Lt] and plan a solution step-by-step!
```

The generated answer comprises a reasoning path represented as a sequence of steps $\{s_i\}_{i=1}^{n}$, where n denotes the number of steps. Formally, this generation can be expressed as:

$$\mathcal{A} = \arg\max p(s_1, \ldots, s_n | Q, L_s, L_t), \quad (1)$$

where $\mathcal{A}$ denotes a specific path generated from the prompted LLM.

Language-specific Solver phase After achieving the planned solution in [Language L_t] in the first phase, we introduce the resolution phase to elicit the model to deliver the final solution. In particular, given the target `[Language` L_t`]`, and the generated planning $\mathcal{A}$, we structure the prompt to engage in resolving target question Q. The model generates the final answer A_t in `[Language` L_t`]` along possible reasoning steps $R = \{r_i\}_{i=1}^{n}$, where n represents the number of steps in the reasoning process delivered by the LLMs. Specifically, we construct the *resolution prompter* as:

```
After understanding, you should act as an
[Language Lt] programmer, and for clarity
at the end, you should format your answer
as 'Die Antwort ist: [num].'
```

Hence, the reasoning paths R are organized into the final reasoning path $\mathcal{R}_t$ for target Language L_t, which can be determined as:

$$\mathcal{R}_t = \arg\max_{R} p(R | P, L_t, Q), \quad (2)$$

where P represents previous generation in $\mathcal{A}$ given the input variables Q, L_s, L_t.

The final answer is determined as:

$$A_t = \arg\max p(f | \mathcal{R}_t), \quad (3)$$

where the model A_t represents the answer generated from all potential reasoning results in f. In conclusion, we evaluate the accuracy by estimating the exact matching between A_t and the target answer.

2.3 Cross-lingual Self-consistent Prompting

Moreover, to align the possible different reasoning paths across different languages, we introduce Self-consistent Cross-PAL ($\mathcal{S}$Cross-PAL) by ensembling the different pathways as proposed in (Wang et al., 2023).

Starting with the fact that a model could deliver distinct reasoning patterns in each specific language, inspired by Wang et al. (2023), we propose $\mathcal{S}$Cross-PAL to integrate reasoning knowledge across different languages (see Figure 1).

Specifically, as described in Section 2.2, during the first step, we prompt the model to understand the problem in different target languages L_t and plan a solution going forward with respective reasoning steps. Hence, in order to select the most consistent reasoning patterns, answers that exhibit a high level of consistency in the inferred generated answers (Y) are selected through a voting mechanism. The final result can be formulated as:

$$\hat{A} = \text{argmax} \sum_{t=1}^{|L|} \sum_{f}^{|Y|} \mathbb{1}\left(A_t = Y\right), \quad (4)$$

where $|L|$ represents the count of target languages, $|Y|$ signifies the count of potential reasoning results Y across all target languages, and $\mathbb{1}\ (X)$ denotes a 0-1 function that returns 0 when X is False and returns 1 when X is True.

3 Experiments

3.1 Data

To observe the mathematical reasoning multilingual[1] abilities of Large Language Models (LLMs), we used GSM8K (Cobbe et al., 2021), and MSVAMP (Chen et al., 2023a).

Multilingual Grade School Math To evaluate the problem-solving abilities in Cross-lingual scenarios, we used the extension proposed by Shi et al. (2022), i.e., Multilingual Grade School Math (MGSM). Initially, Cobbe et al. (2021) proposed a benchmark of mathematical problems in English in GSM8K. Each example has the following structure: a mathematical problem in natural language and a target answer in Arabic number. Shi et al. (2022), in their contribution, i.e., MGSM, selected the first 250 examples from the official list of examples in GSM8K and translated them manually into 11 different languages, maintaining the structure of the input and output.

Multilingual SVAMP Following Shi et al. (2022), Chen et al. (2023a) proposed the multilingual extension of SVAMP (MSVAMP). Patel et al. (2021) similarly structured SVAMP to GSM8K so that the question-and-answer structure is the same as discussed above. However, in contrast to GSM8K, SVAMP has a more significant number of problems but with a lower order of complexity (Patel et al., 2021).

[1]The available languages differ depending on the resources and are listed in Appendix 5.

Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
Single-step											
`Direct` (Qin et al., 2023)	56.0	60.0	62.0	62.0	48.0	61.2	33.6	52.8	7.6	42.2	48.5
`Native-CoT` (Qin et al., 2023)	70.0	59.6	64.4	62.4	54.0	70.4	26.4	64.4	40.0	59.6	57.1
`En-CoT` (Qin et al., 2023)	73.6	63.2	70.0	65.6	55.2	69.6	50.0	60.4	22.0	48.0	57.7
`Translate-En` (Qin et al., 2023)	75.6	71.6	72.4	72.8	69.6	74.4	66.4	66.0	58.0	57.6	68.4
`XLT` (Huang et al., 2023)	81.4	71.8	79.2	80.2	71.2	81.6	64.4	72.8	40.8	69.8	71.3
`Cross-ToT` (Ranaldi et al., 2024c)	**87.0**	78.0	**82.4**	85.6	75.0	84.6	77.0	77.8	**62.0**	70.4	77.0
Double-step											
`CLP` (Qin et al., 2023)	80.0	73.6	79.2	81.6	74.8	82.4	64.8	69.2	38.8	62.0	70.6
`Cross-PAL`	84.2	79.8	82.0	86.8	78.0	84.6	79.0	80.2	59.8	73.6	79.3
+Self-Consistency											
`SCLP` (Qin et al., 2023)	86.8	77.2	82.0	87.6	76.0	84.8	75.2	77.2	52.0	68.0	76.7
`SCross-PAL`	85.8	**81.0**	**84.2**	**88.4**	**79.8**	**86.2**	**79.6**	**81.6**	61.8	**74.0**	**80.3**

Table 1: Accuracies (**`GPT-3.5-turbo`**) on MGSM using the "`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, the "`Translate-En`" prompt where the specific input is translated into English and the answer accordingly is in English. Moreover, CLP and SCLP, as proposed in (Qin et al., 2023), questions in a specific language and answers in different languages. Finally, our **`Cross-PAL`** and **`SCross-PAL`** are explained in Sections 2.2 and 2.3. The best results are reported in bold.

3.2 Experimental Setup

To conduct our study on robust models and have a term of comparison with the work proposed in (Shi et al., 2022; Qin et al., 2023), we use `GPT-3.5`. Furthermore, to show the scalability and effectiveness of our approach on further models, we use Llama3-8, Phi-3, and Llama2-7. We use the last model because it has been employed in previous works (Qin et al., 2023; Ranaldi et al., 2024c). We report in Appendix J details of model versions and parameter configurations. Then, we systematically defined the input prompt as described in Section 2. In each experimental set-up, we modify the appropriate languages L_s, L_t, as shown in Figure 1 for the Chinese.

Following Kojima et al. (2023), we evaluate performance using the accuracy score by computing the string matching between the final answers (see Figure 1 where the final outputs have the form of `"The answer is [num]"`) in specific language and the target values.

4 Main Results

Prompting mechanisms for eliciting Large Language Models (LLMs) in delivering multilingual reasoning answers can be empowered via Cross-lingual Program-Aided Language Models (`Cross-PAL`) that employ a strict in-context structure and aid LLMs in delivering robust reasoning paths across languages. Our approach based on a PAL-inspired prompting mechanism outperforms state-of-the-art in-context learning techniques on Arithmetic Reasoning tasks as shown in Table 1 and Figure 2.

`Cross-PAL` operating in two phases improves the effectiveness of LLMs in understanding and following structured solutions to multilingual reasoning problems. The in-context demonstrations are provided in the understanding (*"understander"* in Figure 1) phase and used as resolution schemes in the resolution phase (*"solver"* in Figure 1). Hence, an alignment mechanism between languages is applied, promoting the reasoning abilities of more robust languages while preserving one of the original questions and consequently allowing proficiency in less robust languages. Finally, Self-consistent `Cross-PAL` (`SCross-PAL`) allows the sampling of different reasoning paths by selecting the most consistent that leads the LLMs to the most accurate path.

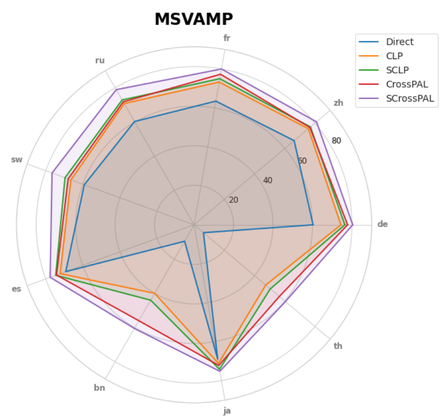

Figure 2: Accuracies (%) on MSVAMP. In Appendix B are reported detailed results.

Our approach outperforms the methods proposed in (Shi et al., 2022) as well as the Cross-lingual Prompting (CLP (Qin et al., 2023) and XLT (Huang et al., 2023)) methods. Having shown that Cross-PaL outperforms previous in-context learning approaches, we would now like to analyse which dynamics emerge between languages (Section 5), the effects of the use of specific language (Section 5.2), the trade-off between the number of languages and the final results (Section 5.3) and finally the operability in smaller models (Section 5.4).

5 Analysis & Discussion

In this section, we explore in depth the impact of program-of-thoughts demonstrations along our Cross-PaL (Section 5.1) discerning between the role of English (in Section 5.2), different languages and in-context demonstrations (Section 5.3) in the evaluation phase. Then, we analyse the scalability of our approach on different models in Section 5.4.

5.1 Cross-PaL multilingual operability

The program-inspired structure of in-context demonstrations positively influences LLMs' multilingual mathematical problem-solving planning. Cross-PaL compared to the previous methods based on single-step prompting (Direct, CoT-based in Table 1), achieves better performances. In addition, Cross-PaL outperforms of 10.9% the CLP-based approach (that uses double-step prompting stages (Qin et al., 2023)). Moreover, introducing self-consistent methods (i.e., *S*Cross-PaL) surpassed the original version (see Cross-PaL and *S*Cross-PaL in Table 1). These results are further confirmed in additional tasks such as MSVAMP in Figure B and on XCOPA (Ponti et al., 2020) in Table 7. The pragmatic nature of in-context demonstrations elicits LLMs to plan problem solutions systematically and encourages alignment between different languages (see example generations in Table 15) by improving the reasoning evolution. Cross-PaL is based on a two-step prompting approach as introduced in Section 2.3, however although on a smaller scale, in Table 2, it can be observed that the benefits of Cross-PaL also emerge when operating with a single-step prompting approach (we reproduced the approach in Section 2.3 using the same strategy on a unique prompt).

However, to better understand the emerging dynamics and the role of each language in the planning and problem-solving phase, we investigate the effect of introducing a more robust language in both English in Section 5.2 and other languages Section 5.3

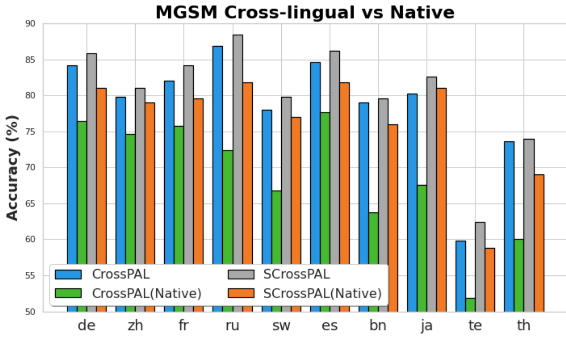

Figure 3: Accuracies (%) on MGSM using Cross-PaL, *S*Cross-PaL, Cross-PaL(Native) and *S*Cross-PaL(Native). The native versions are based on the prompt in Appendix E.

5.2 The English Matter

Earlier works (Wei et al., 2022; Liu et al., 2023) have shown that LLMs are able to deliver multi-step reasoning answers on arithmetic tasks in English and in further languages (Shi et al., 2022) without any fine-tuning phases. Therefore, we analysed the effects of the introduction of English on downstream performance, specifically, the impact of the two phases introduced in Section 2.2 (Figure 1) on the different languages.

Hence, we introduced Cross-PaL(Native) by manipulating the setting proposed in Section 3.2. We replaced the alignment part (the *English* generation part) with the language related to the specific question (see details in Appendix E).

The results obtained in Figure 3 show that the original Cross-PaL outperforms the Cross-PaL(Native). Specifically, there is a substantial difference between the low-resource languages (see Telugu (te) and Swhali(sw), Bangla(bn) and Thai(th)). This result suggests that an alignment phase in a high-resource language (e.g. English) impacts the final reasoning process.

On the other hand, *S*Cross-PaL and *S*Cross-PaL(Native) do not suffer from the lack of English as high-resource languages are present in the path set. However, whether the performance is due to the number of languages or English is unclear. To observe the impact of adding a specific language in Section 5.3, we propose to reduce the number of languages in the presence and absence of English.

Model	Method	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
`GPT-3.5`	*Single-step*	78.8	75.4	79.8	77.6	76.2	81.2	74.6	77.4	55.0	72.0	74.7
	First-step	79.0	75.4	79.8	77.6	80.1	81.6	73.8	77.2	57.4	72.8	75.0
	Double-step	84.2	79.8	82.0	86.8	78.0	84.6	79.0	80.2	59.8	73.6	79.3
`Llama-2`	*Single-step*	49.6	49.4	51.8	52.2	47.6	57.8	24.0	48.2	26.2	43.0	44.8
	First-step	50.4	51.6	52.6	53.0	48.8	59.0	24.8	48.6	27.8	45.0	46.1
	Double-step	54.6	55.0	58.2	57.8	52.6	62.6	29.0	52.0	30.0	47.4	49.9
`Llama-3`	*Single-step*	56.8	56.8	59.6	60.6	52.0	59.8	37.6	52.6	31.8	50.4	51.8
	First-step	56.0	56.4	58.0	58.4	50.8	59.0	37.0	52.0	32.0	50.0	51.0
	Double-step	62.0	59.6	61.2	62.6	58.4	63.0	38.2	56.0	40.4	53.0	55.4
`Phi-3`	*Single-step*	52.6	54.0	57.0	57.8	52.6	58.6	30.2	47.4	34.0	45.2	48.9
	First-step	52.0	53.4	56.2	57.0	51.4	57.4	29.0	47.0	32.6	44.8	47.0
	Double-step	62.0	58.4	60.6	60.2	57.6	62.6	32.8	53.2	39.0	52.8	53.8

Table 2: performances from single-stage prompting (*Single-step*), two-stage prompting (*Double-step*), and only from the first step in the two-stage prompting (*First-step*) by using `Cross`-PAL as presented in Section 2.2

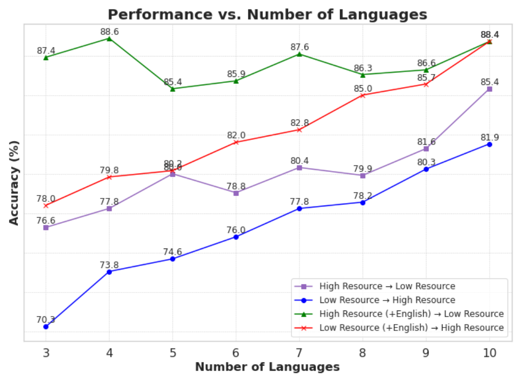

Figure 4: The impact of integrating languages in our *S*`Cross`-PAL on the final performance. Following Table 13, we integrate languages from low-resources to high-resources and vice versa. We also propose the same experiments without the presence of English.

5.3 The Role of the Languages

In *S*`Cross`-PAL, English leads multilingual reasoning on arithmetic tasks, as discussed in Section 5.2. To observe the impact of the number of languages and one specific, i.e., English, we propose two further analyses:

***S*`Cross`-PAL in low-resources scenarios** Integrating more languages into cross-lingual prompting methods (*S*`Cross`-PAL) improves performance. As already observed in (Qin et al., 2023), increasing the number of languages improves downstream performance, as shown in Figure 4 (average performances using the same setting proposed in Section 3.2).

As discussed in (Malkin et al., 2022; Blevins and Zettlemoyer, 2022), the performances of the LLMs are highly correlated with the percentage of pre-training data in each language.

Following the approach proposed in (Qin et al., 2023) and considering language distribution in the widely used multilingual pre-training dataset, which in our case is CommonCrawl (Common Crawl, 2021), we integrated languages in descending and ascending order based on their respective proportions (detailed in Table 13).

Figure 4 indicates that adding more languages in high-resource contexts positively impacts downstream performance. In contrast, performance drops when incorporating low-resource languages increases as the number of languages increases (see low-resource in Table 13). Finally, adding *English* (the most prevalent language in standard pre-training corpora) to the prompting significantly improves performance (see "+ English" in Figure 4).

These findings highlight that the number of languages only partially defines the effectiveness of language integration. The amount of pre-training data for each language, particularly for high-resource languages, plays a strategic function. Balancing multiple languages and considering available resources and impact is crucial.

***S*`Cross`-PAL in bilingual settings** To analyse the effects of English on the *S*`Cross`-PAL approach, we evaluate similar scenarios in low-resource settings and reproduce the same experiments using only two languages. In particular, we used the same setting proposed in Section 3.2 by including an additional path derived from original PAL that is in English (example prompt in Appendix E).

From the results shown in Figure 5 (violet and green bar), using the target English-language tuple does not change in a significant way the performance of high-resource languages. In contrast, low-resource languages achieve significantly higher performances. This second finding reinforces what

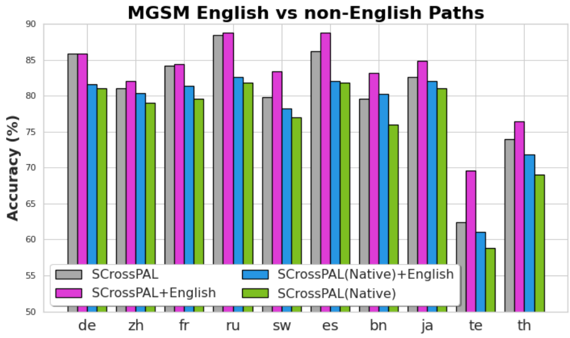

Figure 5: Accuracies (%) on MGSM using *S*Cross-PAL, *S*Cross-PAL(Native), *S*Cross-PAL+*English* and *S*Cross-PAL(Native)+*English*. We define as *English* the original PAL prompt and relative generated reasoning path.

was said earlier about the experiments on prompt compositions.

From Zero- to Few-shot In the original experiments, we used two in-context demonstrations. However, increasing the number of demonstrations does not dramatically improve performance (see Table 3). In contrast, decreasing the number of demonstrations of structured solutions negatively affects performance. In conclusion, a few demonstrations of solutions are necessary to have positive effects and to stimulate models to structure robust solutions.

5.4 Smaller Models

Cross-PAL and *S*Cross-PAL outperform other approaches in open-source models with fewer parameters. Table 4 shows the average scores of Phi-3 and Llama models (Appendix D reports the accuracies of each language). Unlike previous approaches based on in-context natural language rationales, the style of program-based demonstrations is more strict and functional for planning solutions to multilingual mathematical reasoning tasks. This scenario benefits the understanding abilities of various scale models, simplifying the planning problem solutions.

6 Related Work

In-context learning abilities of Large Language Models (LLMs) are based on a series of approaches that elicit models to generate desired planned answers (Brown et al., 2020; Wei et al., 2022; Min et al., 2022). These approaches marked the beginning of the prompting era and were quickly followed by methods to elicit algorithmic and structured reasoning (Roy and Roth, 2015; Ling et al., 2017). Gao et al. (2022) in Program-Aided Language Models (PAL) and parallel Chen et al. (2023b) in Program-of-Thought refined the original idea of Chain-of-Thought (CoT) (Wei et al., 2022) by using structured in-context demonstrations instead of natural language rationales.

The original and derived CoT mechanisms achieved significant success but are limited to generating answers within English. Shi et al. (2022) proposed a multilingual evaluation that Qin et al. (2023) extended to cross-lingual scenarios. Particulary, Qin et al. (2023) introduced a prompting mechanism to handle requests in any language and generate CoT specifically in English. This approach has been proposed both in single-phase, i.e., as a single prompt (CLP) also adopted by (Huang et al., 2023) and multi-phase (SCLP), i.e., characterized by self-consistent prompts that follow the prompting methodology proposed in (Qiao et al., 2023). In our work, we propose a method inspired by the PAL approach and two novel multilingual refinement approaches. In particular, using in-context learning settings (without further fine-tuning), we stimulate generations of thought programs as they are structured and more precise than natural language. Hence, our technique generates cross-lingual structured reasoning paths by providing self-consistent answers. Our work makes the following contributions: *(i)* Proposal of novel PAL-based prompting methods in cross-lingual scenarios characterized by low-resource and high-resource languages. *(ii)* Using arithmetic reasoning tasks to study Cross-PAL multi-step reasoning mechanisms. *(iii)* In-depth study of the reasoning pathways provided by our prompting approach (impact of the number of languages and strongly high-resource languages). *(iv)* Scale-down tests by transferring proposed methods to further LLMs by analyzing performances.

7 Future Work

The study of LLMs' reasoning capabilities in non-English settings is an emerging research domain. Multiple studies have proposed techniques to increase (Ranaldi et al., 2024b; Zhu et al., 2024b; Chen et al., 2024), transfer (Ranaldi and Pucci, 2023), or align (Chai et al., 2024; Zhu et al., 2024a; Ranaldi et al., 2024a; Ranaldi and Pucci, 2024) reasoning capabilities beyond English. Although our contribution has been strongly focused on the benefits of manipulating in-context demonstrations

# of shot- Cross-PAL	de	zh	fr	ru	sw	es	bn	ja	te	th	Avg
0-shot	78.0	76.8	77.6	72.8	60.0	80.2	62.0	69.6	50.0	51.4	67.7 **(-11.6)**
1-shot	82.0	78.0	79.0	80.6	66.8	82.4	67.2	75.0	54.0	63.4	74.8 **(-5.5)**
2-shot (Cross-PAL)	84.2	79.8	82.0	86.8	78.0	84.6	79.0	80.2	59.8	73.6	79.3
3-shot	84.8	80.4	81.8	87.0	78.6	84.4	80.4	82.2	62.2	73.8	79.5 **(+0.3)**
4-shot	85.6	81.4	82.4	87.6	79.8	84.6	81.8	82.0	64.4	75.0	80.3 **(+1.0)**

Table 3: Accuracies (%) on MGSM using zero-shot, one-shot, and three-shot and Cross-PAL (based on two shot in-context demonstrations).

	Model	Direct	CLP	Cross-PAL	SCLP	SCross-PAL
MGSM	**Llama2-7**	42.5	48.3	**49.9**	54.1	**56.3**
	Llama3-8	48.2	53.2	**55.4**	60.6	**62.4**
	Phi-3	43.0	51.3	**54.0**	**57.8**	57.3
MSVAMP	**Llama2-7**	46.8	53.1	**55.0**	58.2	**60.3**
	Llama3-8	51.9	55.7	**57.3**	63.6	**64.4**
	Phi-3	47.0	50.8	**52.5**	61.4	**61.9**

Table 4: Differences in term of accuracies (δ) between Direct and and the Native-based versions (Native-CoT and Native-PAL). Detailed results are provided in Appendix D.

to maximize the understanding and generation capabilities of LLMs, we will be interested in continuing our studies. In particular, we would like to investigate the impacts of further tuning in proposed experimental settings using synthetic data delivered by models of the same family (Ranaldi and Freitas, 2024a) or self-generated (Ranaldi and Freitas, 2024b). In addition, we would like to investigate the impact of non-English reasoning methods in settings related to tasks involving retrieval, logical inference, and other tasks not analysed in this account.

8 Conclusion

In-context *reasoning methods* are effective prompting techniques. However, the imbalance of languages in pre-training data does not always produce robust results. Different state-of-the-art works have proposed multi- and cross-lingual prompting approaches to improve performances obtained across different languages using natural language rationales. In this paper, we propose Cross-PAL, which elicits multi-step reasoning abilities in cross-lingual scenarios. Hence, we elicit models to plan solution problems using a program-like structure. We show the functionality of our Cross-PAL through performance improvements obtained in a multilingual mathematical problem task. Hence, we conducted a series of in-depth analyses to measure the impact of low- and high-resource languages and the inclusion of English. Our contribution aims to propose more robust models that can break down issues arising from language barriers and provide more reliable results.

Limitations

Due to the limitations imposed by the evaluation benchmarks and the cost of the OpenAI API, we conducted tests on limited tasks and different languages, which only scratches the surface of the world's vast array of languages. We tested GPT-based models (closed-source) and several models (open-source) 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 analysed different models in our work, we would like to take a closer look at the performance achieved by language-specific pre-trained models (language-centered). However, at the moment, there are not many open resources comparable in size to those we have analysed. In the future, we hope these models can be readily available to better investigate this phenomenon.

Ethics Statemet

We do not address ethical topics. The data comes from open-source benchmarks, and statistics on language differences in commonly used pre-training data were obtained from official sources.

Acknowledgments

This work is supported in part from UK Research and Innovation under the UK government's Horizon Europe funding guarantee grant number 10039436.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explain the cross-lingual capabilities of english pretrained models.

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.

Linzheng Chai, Jian Yang, Tao Sun, Hongcheng Guo, Jiaheng Liu, Bing Wang, Xiannian Liang, Jiaqi Bai, Tongliang Li, Qiyao Peng, and Zhoujun Li. 2024. xcot: Cross-lingual instruction tuning for cross-lingual chain-of-thought reasoning.

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.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. 2024. Monolingual or multilingual instruction tuning: Which makes a better alpaca. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1347–1356, St. Julian's, Malta. Association for Computational Linguistics.

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

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. *ArXiv*, abs/2110.14168.

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

Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng. 2022. Leveraging knowledge in multilingual commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3237–3246, Dublin, Ireland. Association for Computational Linguistics.

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.

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

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.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4.

Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4903–4915, Seattle, United States. Association for Computational Linguistics.

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.

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.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. 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.

Leonardo Ranaldi and Andre Freitas. 2024a. Aligning large and small language models via chain-of-thought reasoning. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1812–1827, St. Julian's, Malta. Association for Computational Linguistics.

Leonardo Ranaldi and Andrè Freitas. 2024b. Self-refine instruction-tuning for aligning reasoning in language models.

Leonardo Ranaldi and Giulia Pucci. 2023. Does the English matter? elicit cross-lingual abilities of large language models. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 173–183, Singapore. Association for Computational Linguistics.

Leonardo Ranaldi and Giulia Pucci. 2024. When large language models contradict humans? large language models' sycophantic behaviour.

Leonardo Ranaldi, Giulia Pucci, and André Freitas. 2024a. Does the language matter? curriculum learning over neo-Latin languages. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5212–5220, Torino, Italia. ELRA and ICCL.

Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2024b. Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 7961–7973, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, Elena Sofia Ruzzetti, and Fabio Massimo Zanzotto. 2024d. A tree-of-thoughts to broaden multi-step reasoning across languages. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1229–1241, Mexico City, Mexico. Association for Computational Linguistics.

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.

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.

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, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Wenhao Zhu, Shujian Huang, Fei Yuan, Cheng Chen, Jiajun Chen, and Alexandra Birch. 2024a. The power of question translation training in multilingual reasoning: Broadened scope and deepened insights.

Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024b. Question translation training for better multilingual reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 8411–8423, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics.

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

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

B Performances MSVAMP

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	12.6	68.9	20.2	53.7
	CLP	74.4	75.6	73.3	70.8	66.4	72.2	42.3	71.2	47.4	65.7
	SCLP	76.4	77.0	75.0	72.9	69.6	74.6	45.0	74.2	50.4	68.0
	Cross-PAL	76.8	76.3	76.3	71.7	67.8	74.3	56.0	72.2	53.4	69.6
	$\mathcal{S}$Cross-PAL	**80.4**	**81.0**	**78.6**	**76.9**	**74.7**	**77.6**	**60.5**	**75.2**	**59.4**	**74.1**

Table 6: Accuracies (%) on MSVAMP using Cross-PAL, $\mathcal{S}$Cross-PAL and other similar approaches. *we reproduced CLP and SCLP by using the core released by Qin et al. (2023).

C Performance on Symbolic task

Model	Method	et	ht	id	it	qu	sw	ta	th	tr	vi	Avg
GPT-3.5	Direct	90.6	72.0	90.4	95.2	84.6	82.0	59.0	77.6	91.0	83.6	80.2
	CLP	89.6	79.4	94.2	92.8	63.6	84.8	73.4	87.8	91.2	90.8	85.3
	Cross-PAL	91.8	80.6	93.0	96.0	83.4	82.4	77.0	87.4	92.6	91.2	88.2
	SCLP	96.8	90.6	95.2	95.8	85.8	89.8	82.8	83.2	92.8	94.2	95.8
	SCross-CoT	96.4	91.2	96.0	**97.8**	86.6	87.8	84.8	**93.8**	**97.2**	95.6	93.2
Llama-2-7	Direct	39.6	32.5	58.4	55.8	47.2	34.6	47.4	33.2	43.0	59.6	50.4
	CLP	42.0	37.4	59.8	56.6	49.0	38.8	49.2	36.4	45.2	62.8	55.6
	Cross-PAL	44.8	40.4	60.0	56.2	50.2	39.0	49.8	37.2	46.0	63.2	56.8
Llama-3-8	Direct	45.8	37.8	61.0	59.6	49.4	40.0	52.2	46.6	46.8	60.8	56.6
	CLP	48.2	40.0	62.2	59.0	48.8	41.4	52.6	47.2	49.0	61.2	60.4
	Cross-PAL	50.4	41.6	64.0	61.2	50.4	43.0	54.2	48.4	52.2	63.0	62.6
Phi	Direct	43.8	36.2	57.4	55.2	48.6	36.2	50.0	43.8	48.2	56.2	55.4
	CLP	44.0	37.6	57.8	56.4	50.4	38.6	52.4	45.0	44.2	57.0	57.0
	Cross-PAL	46.2	38.4	59.6	56.8	52.0	40.4	53.0	45.8	44.8	58.6	58.6
HUMAN	-	**98.2**	**96.4**	**100.0**	97.0	**94.4**	**98.0**	**98.6**	92.8	96.4	**98.4**	**97.6**

Table 7: Accuracies (%) on symbolic task (XCOPA (Ponti et al., 2020)) using the reasoning methods described in Section 2.2 adapted for textual explainations. ***HUMAN** from (Ponti et al., 2020).

D Performances 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
	`CLP` (Qin et al., 2023)	53.2	53.8	56.0	57.6	**52.8**	59.4	25.6	49.4	29.0	46.0	48.3
	`Cross-PAL`	**54.6**	**55.0**	**58.2**	**57.8**	52.6	**62.6**	**29.0**	**52.0**	**30.0**	**47.4**	**49.9**
							MSVAMP					
`Llama2-7`	`Direct`	51.2	57.3	57.1	51.0	50.9	56.3	10.4	60.2	-	30.1	46.8
	`CLP` (Qin et al., 2023)	52.3	56.2	59.6	58.0	**55.3**	60.4	29.8	65.8	-	41.4	53.1
	`Cross-PAL`	**56.6**	**60.3**	**62.0**	**59.8**	55.0	**62.7**	**35.8**	**66.2**	-	**42.3**	**55.0**

Table 8: Accuracies (%) on MGSM and SVAMP of further models using the reasoning methods described in Section 2.2. We reproduced (Qin et al., 2023) using the released repository.

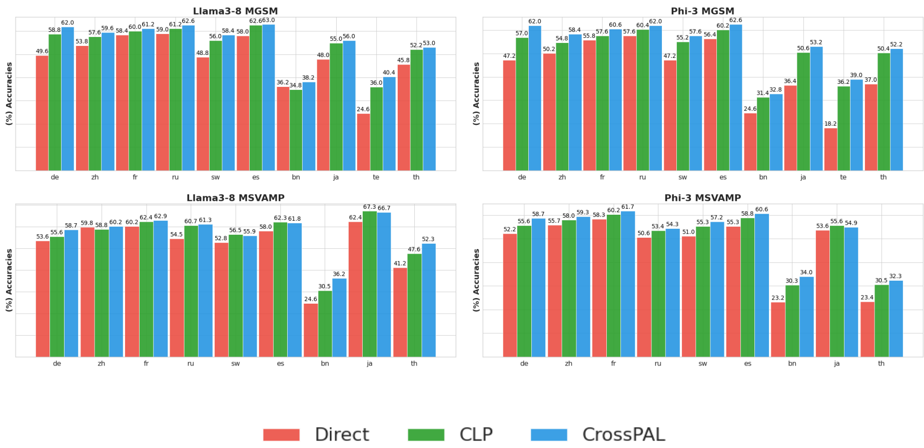

Table 9: Average on MGSM and MSVAMP using `Direct`, `CLP` and `Cross-PAL`. We reproduced (Qin et al., 2023) using the released repository.

E Cross-PAL(Native) prompting approaches (Example in German)

Given the following problems and the program solutions, 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:#Pythonlösung

```
# Roger begann mit 5 Bällen.
tennis_balls = 5
# 2 Dosen, 3 Tennisbälle pro Dose
bought_balls = 2 * 3 tennis balls.
# Die Antwort.
answer = tennis_balls + bought_balls
# Die Antwort ist 11
```

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 11
```

Please, after understanding the structure of previous examples, given
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?
understand the question in **German** and plan a solution step-by-step!

After understanding, you should act as an **German** programmer, and for clarity at the end, you should format your answer as **'Die Antwort ist: [num].'**

F Cross-PAL prompting approaches (Example in Chinese)

Given the following problems and the program solutions, 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:#用Python 解决

```
# 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: 杰森有20 根棒棒糖。他给了丹尼一些棒棒糖。现在杰森有12 个棒棒糖。杰森给了丹尼多少棒棒糖?
A:#用Python 解决

```
# 杰森有20 根棒棒糖。
    jason_lollipops_initial=20
# 杰森现在有12 个棒棒糖。
    jason_lollipops_left=12
# 答案是
    lollipops_given_to_denny=jason_lollipops_initial
-jason_lollipops_left
# 答案是11
```

Please, after understanding the structure of previous examples, given
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?
understand the question in **English** and plan a solution step-by-step!

After understanding, you should act as a **Chinese** programmer, and for clarity at the end, you should format your answer as 答案是：[数字]。

G 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:** 让我们一步步思考

`Translate-CoT`
(Question translation Google API (Qin et al., 2023))

Q: Roger has 5 tennis balls. He bought 2 more cans of tennis balls. [...] How many tennis balls does he have now? **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:** Leah has 32 pieces of chocolate, and her sister has 42. If they ate 35 pieces, how many pieces do they have left? **A:** `Let's think step-by-step!`

`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:** `Let's think step-by-step!`

Table 10: Methods proposed in (Shi et al., 2022) (we reduced the shot, but the original is 6-shot).

`CLIP First-Step`

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

`CLIP Second-Step`

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

Table 11: CLIP (Qin et al., 2023) where the prompt is in two phases: the alignment of the languages and then the solving mechanism for the specific language.

`Cross-ToT`

```
Simulate the collaboration of {n} mathematicians answering a question in their mother tongue: L1, L2, ... and Ln. 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 L1]
Answer: [num].
```

Table 12: `Cross-ToT` (Ranaldi et al., 2024d) that using the Tree-of-Thoughts elicits the model to produce reasoning processes in different languages.

H CommonCrawl distribution

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

I Model and Hyperparameters

In our experimental setting, as introduced in Section 3.2, we propose different LLMs: (i) GPT-3.5 (`gpt-3.5-turbo-0125`), (ii) two models from the Llama family (Touvron et al., 2023): Llama2-7b, Llama2-13b, Llama3-8; (iii) one model of the Phi family: Phi-3 (Abdin et al., 2024) (detailed in Table 14).

In particular, GPT models are used via API, while for the others, we used open-source versions (see detailed versions in Table 14)

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 open-source version of the model on four 48GB NVIDIA RTXA600 GPUs for all experiments performed only in inference.

Finally, the generation temperature 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 have shared the code along with the submission.

J Models Vesions

Model	Version
Llama2-7 Llama3-8(-instruct)	meta-llama/Llama-2-7b meta-llama/Meta-Llama-3-8B-Instruct
Phi-3(-mini-instruct)	microsoft/Phi-3-mini-4k-instruct
GPT-3.5-turbo	OpenAI API (gpt-3.5-turbo-0125)

Table 14: List the versions of the models proposed in this work, which can be found on huggingface.co. We used all the default configurations proposed in the repositories for each model.

K Example of Generations

Cross-PaL

```
(first step) Given the following problems and the program solutions, 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:#用Python 解决
    # 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
(...one more in-context demonstrations as in Table F...)
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Please, after understanding the structure of previous examples, given
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?
understand the question in English and plan a solution step-by-step!
(second step) After understanding, you should act as a Chinese programmer, and for clarity at the end, you should format your answer as 答案是：[数字]。.
```

GPT-3.5-turbo

```
A: # Liya has 32 pieces of chocolate.
    liya_chocolate = 32
     # Her sister has 42 pieces of chocolate.
    sister_chocolate = 42
     # Total chocolate before eating.
    total_chocolate = liya_chocolate + sister_chocolate
     # They ate 35 pieces of chocolate.
    chocolate_eaten = 35
     # Remaining chocolate.
    remaining_chocolate = total_chocolate - chocolate_eaten
  答案是：39 .
```

Llama3-8-Instruct

```
A: As an expert in multilingual understanding in Chinese, I will analyze the question and provide a step-by-step solution. Translation: "Lia has 32 pieces of chocolate, her sister has 42 pieces. If they eat 35 pieces, how many pieces do they have left in total?"
    Calculate the total number of chocolate pieces initially:
    Lia has 32 pieces, and her sister has 42 pieces. To find the total, add the two numbers:
    Total = 32 + 42 = 74
     Calculate the number of chocolate pieces eaten:
    They eat 35 pieces.
     Calculate the number of chocolate pieces left:
     To find the number of chocolate pieces left, subtract the number of eaten pieces from the total:
    Left = Total - Eaten = 74 - 35 = 39
     Lia and her sister have 39 pieces of chocolate left.
剩下= 总数- 吃掉的数= 74 - 35 = 39 .
答案是：39 .
```

Phi-3-mini

```
A: To understand and solve the given problem in both English and with the approach for a Python solution, follow these steps:
Understanding the Problem in English:
Liberty initially has 32 pieces of candy. Her sister has 42 pieces of candy. Together, they eat 35 pieces of candy. We need to find out how many pieces of candy they have left in total.
Plan the Solution:
First, find the total amount of candy Liberty and her sister have initially by adding the candy she has to the amount her sister has. Then, subtract the amount of candy they ate from the total amount they initially had. The result will give us the remaining pieces of candy.
Implement the Solution in Python:
# Liya 32 pieces of candy.
    liya_candy = 32
     # Her sister has 42 pieces of candy.
    sister_candy = 42
     # Total candy initial.
    total_candy = liya_candy + sister_candy
     # Remaining candy after eating.
    candy_eaten = 35
     # Remaining candy.
    remaining_candy = total_candy - candy_eaten
  答案是：39。.
```

Table 15: Example output answers generated by different LLMs prompted via Cross-PaL (§2.2). Generations are structured and guided by program languages.