

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

Anonymous ACL submission

Abstract

In-context learning methods are popular inference strategies where Large Language Models (LLMs) are elicited to solve a task using provided demonstrations without parameter updates. Among these approaches are the reasoning methods, best exemplified by Chain-of-Thought (CoT) and Program-Aided Language Models (PAL), which elicit LLMs to generate reasoning paths, thus promoting accuracy and attracting increasing attention. However, despite the success of these methods, 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. In particular, 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. In addition, we introduce Self-consistent Cross-PAL (SCross-PAL) to ensemble different reasoning paths across languages. Our experimental evaluations show that our method significantly outperforms existing prompting methods, reducing the number of interactions and achieving state-of-the-art performance¹.

1 Introduction

Reasoning methods, best exemplified by Chain-of-Thought (CoT) (Kojima et al., 2023; Wei 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 promoting accuracy and attracting increasing attention. The operation of these methods lies in the LLMs’

abilities to afford tasks by following prompts defined by structured patterns. This phenomenon, known as in-context learning (Brown et al., 2020; Min et al., 2022a; Dong et al., 2023), allows solving tasks without updating the model’s internal parameters using only the input structure.

Since using reasoning methods brings clear benefits to the final performance, a series of earlier works began to investigate whether these findings could also be transferred to languages other than English. Shi et al. (2022) introduced the first multi-lingual benchmark to assess the mathematical reasoning abilities of prompt-based LLMs in different languages, which Huang et al. (2023) exploited to demonstrate the in-context cross-lingual sensitivity of larger LLMs. In parallel, Fang et al. (2022) introduced a pre-translation phase to align and improve the generative reasoning capabilities 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. (2024) 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. These strategies base their efficacy on the generative capabilities of larger-scale LLMs with limited access due to cost and convenience whilst not exploiting the benefits that may arise from the structure of in-context prompting.

In this paper, we propose Cross-lingual Program-Aided Language Models (Cross-PAL), a method for aligning reasoning programs across languages. In particular, 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. Moreover, we introduce Self-consistent Cross-PAL (SCross-PAL), which

¹We have shared the code along with the submission

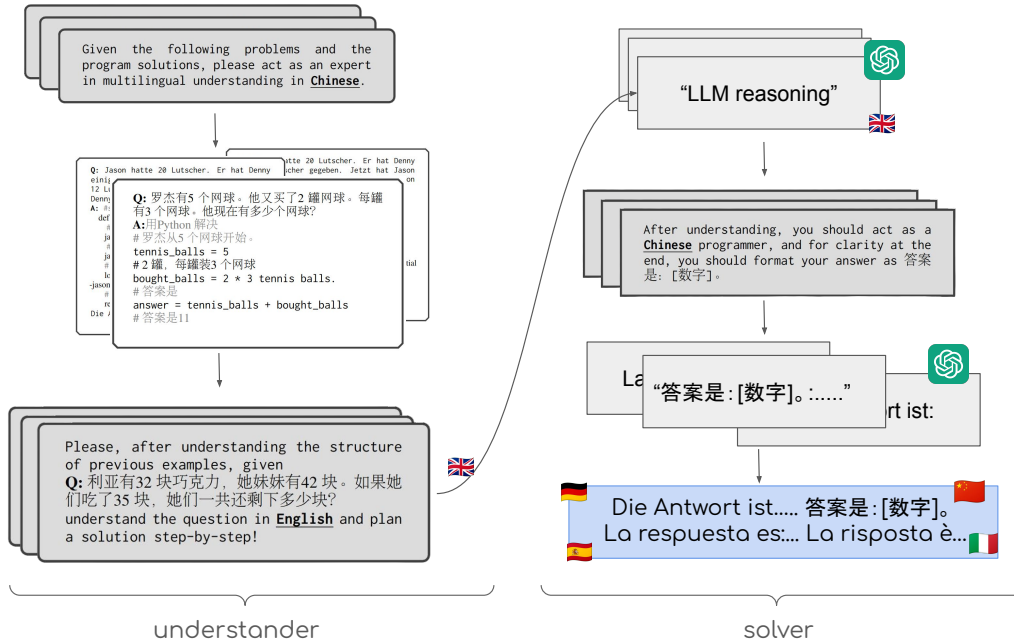


Figure 1: Our Cross-PAL elicits the LLM to generate reasoning programs across different languages. In this example, given separate problems in language L_S , the conducted steps for solving it are: (1) generate a structured planning strategy in English (using in-context demonstrations in L_S), (2) collect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

is inspired by the Self-consistent CoT (Wang et al., 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.

This leads to the main research questions of this paper:

RQ1: What is the role of in-context structured demonstrations, and how much do they impact planning abilities in mathematical reasoning tasks?

RQ2: What effects emerge during the generation process between the different languages, and how does different self-consistent reasoning lead to a more robust final solution?

RQ3: Since previous achievements have been established in high-parameters LLMs. Are the proposed approaches and findings scalable to models with fewer parameters?

To answer these questions, we propose 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 SCross-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 analyze 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 SCross-PAL, which is self-consistent prompting that allows the integration of reasoning paths across different languages. This mechanism enables the align-

137 ment of multilingual abilities by strengthening
138 deficient pathways.

- 139 • Finally, we propose extensive evaluations of
140 the role of each language by discerning be-
141 tween the effects of low-resource and high-
142 resource languages. Hence, we demonstrate
143 that our approaches can effectively improve
144 multilingual performance in mathematical rea-
145 soning tasks and, in contrast to the previous
146 works, achieve stable, robust results in differ-
147 ent scales of LLMs.

148 2 Multi-step Reasoning across Languages

149 To elicit the reasoning abilities of Large Language
150 Models (LLMs) across languages, we propose
151 Cross-PAL, which is a cross-lingual alignment
152 reasoning program, as a solution. In particular,
153 our method goes beyond the previous multilingual
154 prompting strategies introduced in §2.1. Our ap-
155 proach, which takes inspiration from (Gao et al.,
156 2022), uses reasoning programs, also defined as
157 programs-of-thought (Chen et al., 2023b), to struc-
158 ture the generation and elicit the LLMs to deliver
159 structured answers as discussed in § 2.2.

160 2.1 Cross-lingual Chain-of-Thoughts

161 Cross-lingual alignment is a strategic challenge
162 for enabling cross-lingual transfer. Shi et al.
163 (2022) proposed a series of prompts to elicit
164 models to generate CoT answers in specific lan-
165 guage Native-CoT, and in English En-CoT and
166 Translate-CoT (detailed in Table 8). Later,
167 Huang et al. (2023) produced a single prompt tem-
168 plate that showed the in-context cross-lingual sen-
169 sitivity of the more robust LLMs. Qin et al. (2023)
170 extends the previous works method on two phases:
171 Cross-lingual alignment prompt and task-specific
172 solver prompting by using two separate steps, as
173 shown in Table 9, to handle input and output in
174 different languages.

175 Although this second approach overcomes the
176 limitations of previous works that emerged in low-
177 resource languages, it could be more laborious and
178 challenging. Hence, Ranaldi et al. (2023) proposed
179 Cross-ToT, which aims to centralize cross-lingual
180 reasoning in a single pathway by generating self-
181 consistent CoT as reported in Table 10.

182 2.2 Cross-lingual Program-Aided Reasoning

183 To elicit the cross-lingual reasoning ability of
184 LLMs, we operate the structure of the in-context

185 prompt by transferring the PAL (Gao et al., 2022;
186 Chen et al., 2023b) methodology in cross-lingual
187 scenarios (Cross-PAL). Our approach consists of
188 a double-step prompt structured in two steps: first,
189 there is a (i) cross-lingual code-switching under-
190 standing phase (named "understander") followed
191 by a (ii) language-specific resolution phase (named
192 "solver").

193 **Cross-lingual Understander** To elicit LLMs in
194 understanding the provided problem and planning
195 a solution by using in-context structured demon-
196 strations in a specific language "[Language L_s]",
197 we plan the prompt in the following way:

```
198 Given the following problems and the  
199 program solutions, please act as an expert  
200 in multilingual understanding in [Language  
201  $L_s$ ].
```

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

```
212 Q: Jason hatte 20 Lutscher. Er hat Denny  
213 einige Lutscher gegeben. Jetzt hat Jason  
214 12 Lutscher. Wie viele Lutscher hat Jason  
215 Denny gegeben?  
216 A: #solution in Python  
217 def solution():  
218     # Jason hatte 20 Lutscher.  
219     jason_lollipops_initial=20  
220     # Jetzt hat Jason 12 Lutscher.  
221     jason_lollipops_left=12  
222     # Lutscher werden Denny  
223     lollipops_given_to_denny=jason_lollipops_initial  
224     -jason_lollipops_left  
225     # Die Antwort ist 11
```

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

```
230 Please, after understanding the structure  
231 of previous examples, given  
232 Q: [question Q (in  $L_s$ )]  
233 understand the question in [Language  $L_t$ ]  
234 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. In a formal way, this generation can be expressed as:

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

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 L_t] 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 generates the answer A_t chosen 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 (SCross-PAL) by ensembling the different pathways as proposed in (Wang et al., 2023).

Starting by the fact that a model could deliver various reasoning patterns in accordance to a specific language, inspired by Wang et al. (2023), we

propose SCross-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:

$$\hat{A} = \arg \max \sum_{t=1}^{|L|} \sum_f^{|Y|} \mathbb{1}(A_t = Y), \quad (4)$$

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 arithmetic reasoning cross-lingual abilities of Large Language Models (LLMs), we used GSM8K (Cobbe et al., 2021), and MSVAMP (Chen et al., 2023a). The available languages differ depending on the resources reported in Appendix 3.

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) structured SVAMP in a similar manner to GSM8K so that the question-and-answer structure is the same as discussed above. However, in contrast to GSM8K, SVAMP has a larger number of

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., 2024)	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.

problems but with a lower order of complexity (Patel et al., 2021).

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 (the last because it has been tested in previous works (Qin et al., 2023; Ranaldi et al., 2024)). 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 German.

Following Wei et al. (2022); Kojima et al. (2023), we evaluate performance using the accuracy score. In particular, we compute 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 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 Arith-

metic Reasoning tasks as shown in Table 1 and Figure 2.

In particular, 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.

Our approach outperforms the methods proposed in (Shi et al., 2022) that are yet surpassed by the Cross-lingual Prompting (CLP (Qin et al., 2023) and XLT (Huang et al., 2023)). However, although Cross-PAL outperform previous in-context learning approaches, it is necessary to analyze which dynamics emerge between languages (Section 5), whether the introduction of English can increase performance (Section 5.2), the trade-off between the number of languages and the final results (Section 5.3).

5 Analysis

In this section, we explore the impact of program-of-thoughts demonstrations along our Cross-PAL

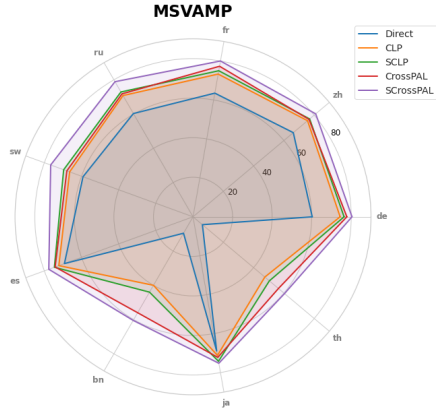


Figure 2: Accuracies (%) on MSVAMP. In Appendix B are reported detailed results.

(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 analyze the scalability of our approach on different models in Section 5.4.

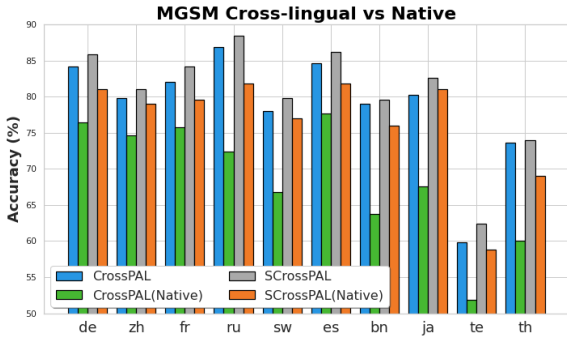


Figure 3: Accuracies (%) on MGSM using CrossPAL, SCrossPAL, CrossPAL(Native) and SCrossPAL(Native). The native versions are based on the prompt in Appendix E.

5.1 Cross-PAL works across the languages

The structure of in-context demonstrations positively influences LLMs’ multilingual mathematical problem-solving planning. Our Cross-PAL outperforms previous methods. In comparison with the previous methods based on single-step prompting (Direct, CoT-based in Table 1), improvements can be observed. In addition, significant average improvements emerged compared to the double-step prompting approaches, i.e., CLP-based (Qin et al., 2023). Moreover, introducing self-consistent methods (i.e., SCross-PAL) has surpassed the original version (see Cross-PAL and SCross-PAL in Table

1). These results are further confirmed in additional tasks such as MSVAMP in Figure B.

The pragmatic nature of in-context demonstrations elicits LLMs to plan problem solutions systematically and encourages alignment between different languages (see generations in Table 14) by improving the reasoning evolution (discussed in Appendix H). 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

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 analyzed 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 Swahili(sw), Bangla(bn) and Thai(th)). This result suggests that an alignment phase in a robust language, in English, impacts the final reasoning process.

On the other hand, SCross-PAL and SCross-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.

5.3 The Role of the Languages

In SCross-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:

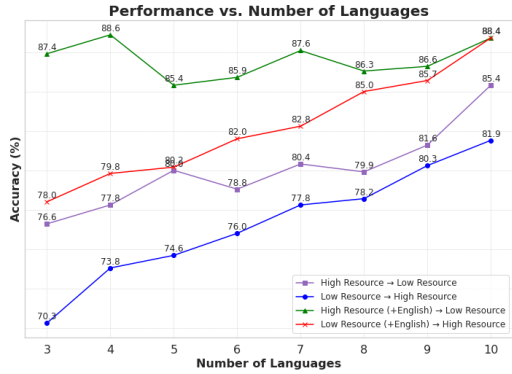


Figure 4: The impact of integrating languages in our *SCross-PAL* on the final performance. Following Table 12, we integrate languages from low-resources to high-resources and vice versa. We also propose the same experiments without the presence of English.

SCross-PAL in low-resources scenarios Integrating more languages into cross-lingual prompting methods (*SCross-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 Large Language Models 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 12).

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 12). Finally, adding *English* (the prevalent rate in standard pre-training corpora) to the prompting significantly improves performance (see "+ English" in Figure 4).

These findings highlight that the number of integrated 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 func-

tion. Balancing multiple languages and considering available resources and impact is crucial.

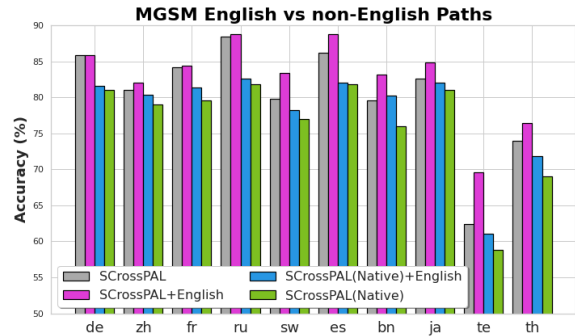


Figure 5: Accuracies (%) on MGSM using *SCross-PAL*, *SCross-PAL(Native)*, *SCross-PAL+English* and *SCross-PAL(Native)+English*. We define as *English* the original PAL prompt and relative generated reasoning path.

SCross-PAL in bilingual settings To analyze the effects of English on *SCross-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 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 5). 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 *SCross-PAL* outperform other approaches in open-source models with fewer parameters. Table 2 shows the average scores of Phi-3 and Llama models (Appendix D reports the accuracies

Model		Direct	CLP	Cross-PAL	SCLP	$S_{\text{Cross-PAL}}$
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-4	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-4	47.0	50.8	52.5	61.4	61.9

Table 2: Differences in term of accuracies (δ) between Direct and and the Native-based versions (Native-CoT and Native-PAL) in Appendix D detailed results.

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 emerged in Large Language Models (LLMs) (Brown et al., 2020; Wei et al., 2022; Min et al., 2022b) are based on a series of approaches that elicit models to generate desired planned answers. These approaches kept the beginning of the prompting era and reached the age of the intermediate steps in algorithmic and structured reasoning (Roy and Roth, 2015; Ling et al., 2017). Nevertheless, earlier works questioned the efficacy of few-shot methods for improving the prompting phase and performances. 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 that are less ambiguous than natural language rationales.

The traditional and derivated CoT mechanisms achieved significant success but are determined to generate answers within a single language (i.e., English). Shi et al. (2022) proposed a multilingual evaluation that Qin et al. (2023) extended to cross-lingual scenarios. In particular, Qin et al. (2023) proposed a prompt mechanism to handle

requests in any language and generate CoT specifically in English. This approach, which in our construct we defined as Cross-CoT, has been proposed both single-phase, i.e., as a single prompt (CLP) also adopted by (Huang et al., 2023) and multi-phase (CLPS), 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 (Li et al., 2024; Zhu et al., 2024). 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 and reducing reasoning steps. Our work goes beyond the following ways: (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-up test by transferring proposed methods to further LLMs by analyzing performances.

7 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, a prompting method that 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.

583 Limitations

584 Due to the limitations imposed by the evaluation
585 benchmarks and the cost of the OpenAI API, we
586 conducted tests on two tasks and different lan-
587 guages, which only scratches the surface of the
588 world’s vast array of languages. Finally, we tested
589 the effectiveness of our method on GPT-based
590 models (closed-source) and several models (open-
591 source). In the future, it will be appropriate to
592 study the generality of our model compared to other
593 closed-source Large Language Models.

594 Finally, although we have considered and ana-
595 lyzed different models in our work, we would like
596 to take a closer look at the performance achieved by
597 language-specific pre-trained models (better known
598 as language-centered). However, at the moment,
599 there are not many open resources comparable in
600 size to those we have analyzed. In the future, we
601 hope these models can be readily available to better
602 investigate this phenomenon.

603 Ethics Statemet

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

610 References

611 Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan,
612 Jyoti Aneja, Ahmed Awadallah, Hany Awadalla,
613 Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jian-
614 min Bao, Harkirat Behl, Alon Benhaim, Misha
615 Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai,
616 Martin Cai, Caio César Teodoro Mendes, Weizhu
617 Chen, Vishrav Chaudhary, Dong Chen, Dongdong
618 Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra,
619 Xiyang Dai, Allie Del Giorno, Gustavo de Rosa,
620 Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan
621 Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg,
622 Abhishek Goswami, Suriya Gunasekar, Emman
623 Haider, Junheng Hao, Russell J. Hewett, Jamie
624 Huynh, Mojan Javaheripi, Xin Jin, Piero Kauff-
625 mann, Nikos Karampatziakis, Dongwoo Kim, Ma-
626 houd Khademi, Lev Kurilenko, James R. Lee, Yin Tat
627 Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Li-
628 den, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin,
629 Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola,
630 Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon
631 Norick, Barun Patra, Daniel Perez-Becker, Thomas
632 Portet, Reid Pryzant, Heyang Qin, Marko Radmi-
633 lac, Corby Rosset, Sambudha Roy, Olatunji Ruwase,
634 Olli Saarikivi, Amin Saied, Adil Salim, Michael San-
635 tacroce, Shital Shah, Ning Shang, Hiteshi Sharma,

636 Swadheen Shukla, Xia Song, Masahiro Tanaka, An-
637 drea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang,
638 Yu Wang, Rachel Ward, Guanhua Wang, Philipp
639 Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can
640 Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang,
641 Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu,
642 Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jian-
643 wen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang,
644 Yunan Zhang, and Xiren Zhou. 2024. *Phi-3 technical
645 report: A highly capable language model locally on
646 your phone.*

Terra Blevins and Luke Zettlemoyer. 2022. *Language
647 contamination helps explain the cross-lingual capa-
648 bilities of english pretrained models.* 649

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie
650 Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind
651 Neelakantan, Pranav Shyam, Girish Sastry, Amanda
652 Askell, Sandhini Agarwal, Ariel Herbert-Voss,
653 Gretchen Krueger, Tom Henighan, Rewon Child,
654 Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu,
655 Clemens Winter, Christopher Hesse, Mark Chen, Eric
656 Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess,
657 Jack Clark, Christopher Berner, Sam McCandlish,
658 Alec Radford, Ilya Sutskever, and Dario Amodei.
659 2020. *Language models are few-shot learners.* 660

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong,
661 Yangqiu Song, Dongmei Zhang, and Jia Li. 2023a.
662 *Breaking language barriers in multilingual mathem-
663 atical reasoning: Insights and observations.* 664

Wenhu Chen, Xueguang Ma, Xinyi Wang, and
665 William W. Cohen. 2023b. *Program of thoughts
666 prompting: Disentangling computation from reason-
667 ing for numerical reasoning tasks.* 668

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
669 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
670 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
671 Nakano, Christopher Hesse, and John Schulman.
672 2021. *Training verifiers to solve math word prob-
673 lems.* 674

Common Crawl. 2021. *Common crawl 2021.* Web.
675 Accessed: 2023-12-12. 676

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong
677 Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and
678 Zhifang Sui. 2023. *A survey on in-context learning.* 679

Yuwei Fang, Shuohang Wang, Yichong Xu, Ruochen
680 Xu, Siqi Sun, Chenguang Zhu, and Michael Zeng.
681 2022. *Leveraging knowledge in multilingual com-
682 mon-sense reasoning.* In *Findings of the Association
683 for Computational Linguistics: ACL 2022*, pages
684 3237–3246, Dublin, Ireland. Association for Compu-
685 tational Linguistics. 686

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon,
687 Pengfei Liu, Yiming Yang, Jamie Callan, and Gra-
688 ham Neubig. 2022. *Pal: Program-aided language
689 models.* *arXiv preprint arXiv:2211.10435.* 690

691	Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Roscoe: A suite of metrics for scoring step-by-step reasoning.	745
692		746
693		747
694		748
695	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.	749
696		750
697		751
698		752
699		
700	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners.	753
701		754
702		755
703		756
704	Bryan Li, Tamer Alkhouli, Daniele Bonadiman, Nikolaos Pappas, and Saab Mansour. 2024. Eliciting better multilingual structured reasoning from llms through code.	757
705		758
706		759
707		760
708	Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 158–167, Vancouver, Canada. Association for Computational Linguistics.	761
709		762
710		763
711		764
712		765
713		766
714	Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023. Evaluating the logical reasoning ability of chatgpt and gpt-4.	767
715		768
716		769
717		770
718	Dan Malkin, Tomasz Limisiewicz, and Gabriel Stanovsky. 2022. A balanced data approach for evaluating cross-lingual transfer: Mapping the linguistic blood bank. In <i>Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 4903–4915, Seattle, United States. Association for Computational Linguistics.	771
719		772
720		773
721		774
722		775
723		776
724		777
725	Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Rethinking the role of demonstrations: What makes in-context learning work?	778
726		779
727		780
728		781
729	Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing</i> , pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.	782
730		783
731		784
732		785
733		786
734		787
735		788
736		789
737	OpenAI. 2023. Gpt-4 technical report.	790
738		791
739	Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In <i>Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies</i> , pages 2080–2094, Online. Association for Computational Linguistics.	792
740		793
741		794
742		795
743		796
744		797
	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 <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.	798
		799
		800
		801
	Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages.	802
		803
	Leonardo Ranaldi, Giulia Pucci, and Andre Freitas. 2023. Empowering cross-lingual abilities of instruction-tuned large language models by translation-following demonstrations.	804
		805
	Leonardo Ranaldi, Giulia Pucci, Federico Ranaldi, Elena Sofia Ruzzetti, and Fabio Massimo Zanzotto. 2024. Empowering multi-step reasoning across languages via tree-of-thoughts.	806
		807
	Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In <i>Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing</i> , pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.	808
		809
	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.	810
		811
	Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutika 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, Madsen 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, Rishi 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.	812
		813
	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.	814
		815

802 Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel,
803 Barret Zoph, Sebastian Borgeaud, Dani Yogatama,
804 Maarten Bosma, Denny Zhou, Donald Metzler, Ed H.
805 Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy
806 Liang, Jeff Dean, and William Fedus. 2022. [Emergent abilities of large language models](#).
807

808 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten
809 Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and
810 Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#).
811

812 Wenhao Zhu, Shujian Huang, Fei Yuan, Cheng Chen,
813 Jiajun Chen, and Alexandra Birch. 2024. [The power of question translation training in multilingual reasoning: Broadened scope and deepened insights](#).
814
815

A Proposed Task

Dataset	Task	Languages	#Languages
MGSM	mathematical reasoning	Bengali (bn), Chinese (zh), French (fr), Thai (th)	10
		German (de), Japanese (jp), Russian (ru), Telugu (te)	
		Spanish (es), Swahili (sw)	
MSVAMP	mathematical reasoning	Bengali (be), Chinese (zh), French (fr), Thai (th)	9
		German (de), Japanese (jp), Russian (ru)	
		Spanish (es), Swahili (sw)	

Table 3: 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
	SCross-PAL	80.4	81.0	78.6	76.9	74.7	77.6	60.5	75.2	59.4	74.1

Table 4: Accuracies (%) on MSVAMP using Cross-PAL, SCross-PAL and other similar approaches. *we reproduced CLP and SCLP by using the core released by [Qin et al. \(2023\)](#).

C Other Results

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

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 6: 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.

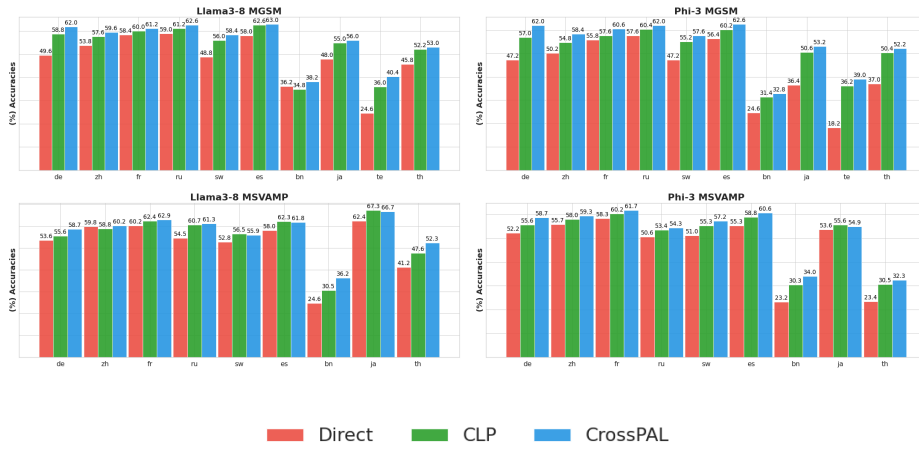


Table 7: Average on MGSM and MSVAMP using Direct, CLP and Cross-PAL. We reproduced (Qin et al., 2023) using the released repository.

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 [Language L_t] 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

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 8: 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 a specific language, the following prompts are Direct, Native-CoT (without additional languages), and En-CoT, the original question in a 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 9: CLIP (Qin et al., 2023) where the prompt is split into two phases: the alignment of the different 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: L_1, L_2, \dots 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 10: Cross-ToT prompting (Ranaldi et al., 2023) that using the Tree-of-Thoughts method elicits the model to produce reasoning processes in different languages.

H Reasoning Evolution

H.1 Chain-of-Thought Quality Scoring Implementation

The ROSCOE framework (Golovneva et al., 2023) incorporates multiple chain-of-thought quality metrics, with the reasoning alignment vector α that is

$$r_{align}(h \rightarrow s) = \{\alpha_1, \alpha_2, \dots, \alpha_N\} \in [0, 1]^N \quad (5)$$

from the N -step hypothesis $h = \{h_i\}_{i=1}^N$ to the source input s of length T , where α_i are defined as:

$$r_{align}(h_i \rightarrow s) = \frac{1 + \max_{j=1}^T \cos(h_i, s_j)}{2}$$

Faithfulness score The Faithfulness (F) score is calculated based on the alignment between the hypothesis steps h and the source sentences s . It represents the average reasoning alignment score over the steps of reasoning:

$$F = \frac{1}{N} \sum_{i=1}^N r_{align}(h_i \rightarrow s) \quad (6)$$

The Faithfulness score serves as a measure to assess whether the model misconstrued the problem in the statement or if the reasoning chain is characterized by ambiguity, unimportance, or the misuse of information.

Informativness Informativness-Step (Info-Step) measures the utilization of facts from the original text s in the reasoning steps h :

$$InfoStep = \frac{1}{2T} \sum_{t=1}^T r_{align}(s_t \rightarrow h) + \frac{1}{2} F \quad (7)$$

Info-Step assigns a higher score to reasoning steps that strongly align with the source, showing the capacity to which the generated hypothesis includes the information from the source. Conversely, a lower Info-Step score means reasoning steps unrelated to the source sentences or overlooking the provided information in the context.

Informativness Chain Like the Info-Step metric, the InformativnessChain (Info-Chain) metric estimates the degree of concordance between the hypothesis chain and the source. The calculation is as follows:

$$InfoChain = \frac{1 + \cos(h, s)}{2} \quad (8)$$

Missing Step The Missing Step (Miss-Step) metric is introduced to estimate any significant lacking steps, which examines the alignment between the reference reasoning text $r = \{r_i\}^K$ and the hypothesis h . A miss-step is needed to meticulously assess each step in the reference and verify the existence of a similar step in the hypothesis. The metric is computed as:

$$\text{Miss-Step} = \min_{i=1}^K(\text{r-align}(r_i, h)). \quad (9)$$

H.2 Application of RSCORE

We use the framework ROSCOE (Golovneva et al., 2023) to investigate why our approach works. Hence, we evaluate the quality of the reasoning paths (implementation described in Appendix H). As shown in Figure 11, our approach delivers reasoning with higher faithfulness, exhibiting better consistency with key steps during the reasoning process. Specifically, the faithfulness score increased by 4.5 points, indicating that the model better understood the problem statement and ensured a transparent inference chain without generating irrelevant or misused information. Furthermore, we observe improvements in the Informativeness metrics for ‘‘Step’’ and ‘‘Chain’’. It suggests that the models’ reasoning, behind the alignment, could provide more well-grounded inference steps.

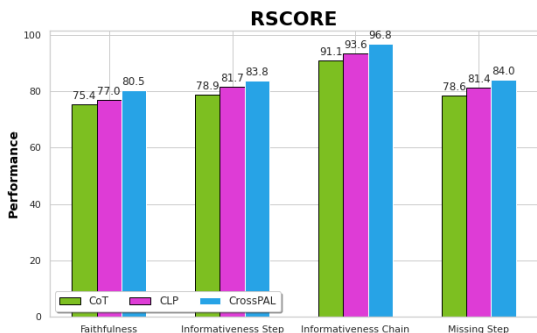


Table 11: The analysis of reasoning quality between GPT-3.5 (Native-CoT), CLP in (Qin et al., 2023) and Cross-PAL

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).

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 13).

In particular, GPT models are used via API, while for the others, we used open-source versions (see detailed versions in Table 13)

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 RTX600 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 Versions

Model	Version
Llama2-7	meta-llama/Llama-2-7b
Llama3-8(-instruct)	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 13: 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 14: Example output answers generated by different LLMs prompted via Cross-PAL (Section 2.2). Generations are structured and guided by program languages.