Unlocking the Capabilities of Thought: A Reasoning Boundary Framework to Quantify and Optimize Chain-of-Thought

Qiguang Chen[†] **Libo Qin**[‡*] **Jiaqi Wang**[◇] **Jinxuan Zhou**[‡] **Wanxiang Che**[†*]
[†] Research Center for Social Computing and Information Retrieval
[†] Harbin Institute of Technology
[‡] School of Computer Science and Engineering, Central South University
[◇] The Chinese University of Hong Kong
{qgchen,car}@ir.hit.edu.cn, lbqin@csu.edu.cn

Abstract

Chain-of-Thought (CoT) reasoning has emerged as a promising approach for enhancing the performance of large language models (LLMs) on complex reasoning tasks. Recently, a series of studies attempt to explain the mechanisms underlying CoT, aiming to deepen the understanding of its efficacy. Nevertheless, the existing research faces two major challenges: (1) *a lack of quantitative metrics to assess CoT capabilities* and (2) *a dearth of guidance on optimizing CoT performance*. Motivated by this, in this work, we introduce a novel reasoning boundary framework (RBF) to address these challenges. To solve the lack of quantification, we first define a reasoning boundary (RB) to quantify the upper-bound of CoT and establish a combination law for RB, enabling a practical quantitative approach applicable to various real-world CoT tasks. To address the lack of optimization, we propose three categories of RBs. We further optimize these categories with combination laws focused on RB promotion and reasoning path optimization for CoT improvement. Through extensive experiments on 27 models and 5 tasks, the study validates the existence and rationality of the proposed framework. Furthermore, it explains the effectiveness of 10 CoT strategies and guides optimization from two perspectives. We hope this work can provide a comprehensive understanding of the boundaries and optimization strategies for reasoning in LLMs. Our code and data are available at https://github.com/LightChen233/reasoning-boundary.

1 Introduction

In recent years, Large Language Models (LLMs) have demonstrated increasing capabilities and applications across various tasks [Zhao et al., 2023, Chang et al., 2023, Pan et al., 2023, Qin et al., 2024a]. Notably, advanced LLMs, such as GPT [Brown et al., 2020, OpenAI, 2022, 2023], PaLM [Anil et al., 2023] and LlaMa [Touvron et al., 2023a,b, Meta, 2024] series have demonstrated emergent capabilities, particularly like Chain-of-Thought (CoT) [Nye et al., 2022, Wei et al., 2022]. This methodology enables models to verbalize step-by-step reasoning, thereby enhancing prediction accuracy by basing decisions on the logical rationale [Wei et al., 2022, Kojima et al., 2022, Hu et al., 2024, Qin et al., 2023, Zhuang et al., 2023, Chen et al., 2024a].

Recently, some research in the literature has begun to investigate the mechanism of CoT to enhance the understanding of its operational nature. To this end, Madaan et al. [2023] and Wang et al. [2023a]

[*] Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

first give a qualitative boundary conclusion through a large number of experiments on the natural language planning capability: The CoT is limited by the reasoning logic in the context demonstrations. Bi et al. [2024] investigate these boundaries on the code planning capability, by training LLMs on CoT samples of varying difficulties. It demonstrates LLMs are unable to learn or effectively manage tasks that exceed a certain complexity upper-bound. To delve deeper into potential constraints of CoT, Feng et al. [2024] develop a theoretical framework on the single-step calculation capability, suggesting that there is an upper-bound of model performance dependent on the length of input in single-step reasoning processes. Although existing research has made some progress, where the boundaries of CoT lie and how these boundaries affect the performance of CoT are still unresolved questions. Specifically, the existing work still faces two major challenges:

- **Lacking quantification metrics for CoT:** Current research primarily relies on qualitative assessments of CoT performance, which leads to the absence of quantitative metrics. It hinders the ability to objectively compare different CoT approaches and establish a definitive upper-bound for CoT capabilities.
- **Lacking optimization guidance for CoT:** While current research prioritizes understanding the mechanisms underlying CoT reasoning, there is a dearth of guidance on optimizing CoT performance. This gap hinders the transformation of CoT research into actionable strategies for enhancing model capabilities.

Motivated by this, in this work, we introduce a reasoning boundary framework (RBF) to thoroughly examine and optimize the boundaries of current LLMs. Specifically, to address the quantification challenge, we propose a new concept, named reasoning boundary (RB) to quantify the upper-bound on task-specific reasoning complexity within a model. Furthermore, to explore more practical scenarios, we present the combination law of RBs to generalize the RB for quantification in more real and complex scenarios. To address the CoT optimization challenge, we propose and analyze three reasoning boundary intervals, guiding optimization through improved RB and optimized reasoning paths based on the combination law, which achieves state-of-the-art performance in our proposed benchmark. We extensively validate the efficacy of our framework across 27 models and 5 tasks: arithmetic computing, mathematical reasoning, multi-hop question answering, and multilingual mathematical reasoning.

Our main contributions are as follows:

- To the best of our knowledge, this is the first work to propose a reasoning boundary framework (RBF) to quantify the upper-bound of CoT. Furthermore, we establish the combination law of RB as the weighted harmonic mean of fundamental RBs to address practical CoT tasks.
- To solve the lack of CoT optimization, we define three categories of RBs. Based on the combination law and the nature of these RBs, we effectively improve the existing CoT strategies by RB promotion and reasoning path optimization.
- We validate the existence and rationality of our framework on 27 models and 5 CoT tasks. Furthermore, we explain the optimal performance from two optimization perspectives in numerous CoT strategies. We consider both optimal perspectives and propose a minimum acceptable reasoning path (MARP) prompting to achieve state-of-the-art performance.

2 Quantification Methodology

2.1 Reasoning Boundary

In order to quantify the capacity for complex reasoning in LLMs, we introduce an upper-bound concept termed reasoning boundary (RB), which formally defines the degree of ease that an LLM can handle within a specific reasoning process. In simpler terms, as shown in Figure 1 (a), RB reflects the limit beyond which a model's accuracy significantly degrades. Mathematically, RB is defined for a model m and a task t as the maximum of problem difficulty d at which the model's accuracy reaches a predefined threshold K_1:

$$\mathcal{B}_{Acc=K_1}(t|m) = \sup_{d}\{d|Acc(t|d,m) \leq K_1\}, \tag{1}$$

where $Acc(t|d,m)$ represents the accuracy of the model's accuracy on task t with difficulty d. Difficulty can be measured by factors like the number of reasoning steps or computational complexity. For brevity, we denote RB as $\mathcal{B}(t|m)$ in subsequent sections.

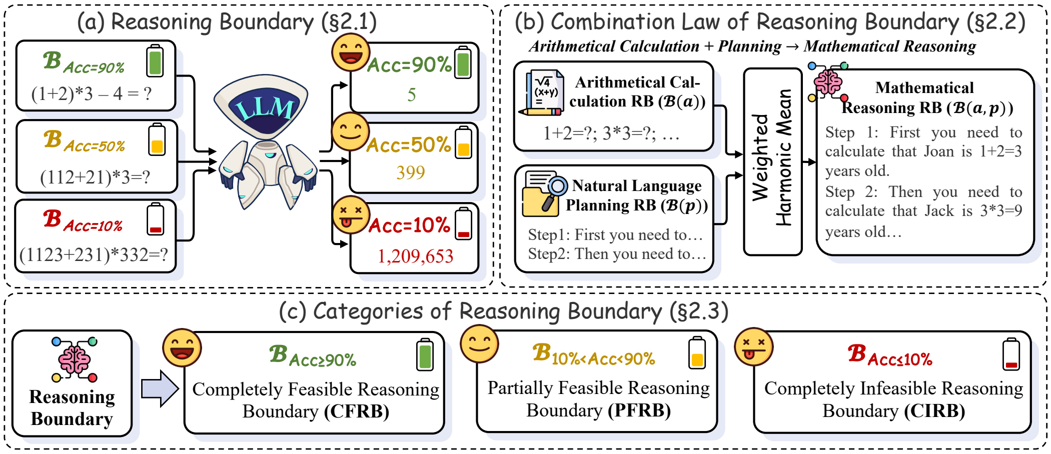

(a) Reasoning Boundary (§2.1)
$\mathcal{B}_{Acc=90\%}$
(1+2)*3 – 4 = ?
$\mathcal{B}_{Acc=50\%}$
(112+21)*3=?
$\mathcal{B}_{Acc=10\%}$
(1123+231)*332=?
LLM
Acc=90%
5
Acc=50%
399
Acc=10%
1,209,653
(b) Combination Law of Reasoning Boundary (§2.2)
Arithmetical Calculation + Planning → Mathematical Reasoning
Arithmetical Calculation RB ($\mathcal{B}(a)$)
1+2=?; 3*3=?; …
Natural Language Planning RB ($\mathcal{B}(p)$)
Step1: First you need to…
Step2: Then you need to…
Weighted Harmonic Mean
Mathematical Reasoning RB ($\mathcal{B}(a,p)$)
Step 1: First you need to calculate that Joan is 1+2=3 years old.
Step 2: Then you need to calculate that Jack is 3*3=9 years old…
(c) Categories of Reasoning Boundary (§2.3)
Reasoning Boundary
$\mathcal{B}_{Acc\geq 90\%}$
Completely Feasible Reasoning Boundary (CFRB)
$\mathcal{B}_{10\%<Acc<90\%}$
Partially Feasible Reasoning Boundary (PFRB)
$\mathcal{B}_{Acc\leq 10\%}$
Completely Infeasible Reasoning Boundary (CIRB)

Figure 1: Overview of the introduced concepts.

Conclusion: The reasoning boundary for a model is defined by its ability to achieve a specific accuracy for a given task difficulty.

2.2 Combination Law of Reasoning Boundary

In practical scenarios, models often require the integration of multiple capabilities to address a single task effectively. To quantify how a large language model can be boosted by the cooperation of multiple capabilities through the CoT mechanism, we introduce the "*Combination Law of RB*", giving a concrete formula of the upper-bound of the CoT. The law estimates the unified reasoning boundary $\mathcal{B}_{\text{Acc}=K_1}(t_1, t_2, \ldots, t_n|m)$ for n tasks within a model m, which is formulated as:

$$\mathcal{B}_{\text{Acc}=K_1}(t_1, t_2, \ldots, t_n|m) \approx \frac{1}{\sum_{i=1}^{n} \frac{N_i}{\mathcal{B}_{\text{Acc}=K_1}(t_i|m) - b_i}}, \tag{2}$$

where $\mathcal{B}_{\text{Acc}=K_1}(t_i|m)$ denotes the reasoning boundary of model m for task t_i. N_i, and b_i are scaling factors, which are only affected by the related task. As shown in Figure 1 (b), Equation (2) provides a mathematical formula to estimate the combined RBs from the independent ones, enabling deeper insights into model behavior for intricate tasks. See Appendix A for detailed mathematical analysis.

Furthermore, the combination law for reasoning boundary demonstrates favorable theoretical properties, with broad applicability across diverse scenarios and flexibility in accommodating various boundary segmentation methods. For detailed practical application, please refer to Appendix B.

Conclusion: The combination law of reasoning boundary satisfies the weighted harmonic average of each basic reasoning boundary.

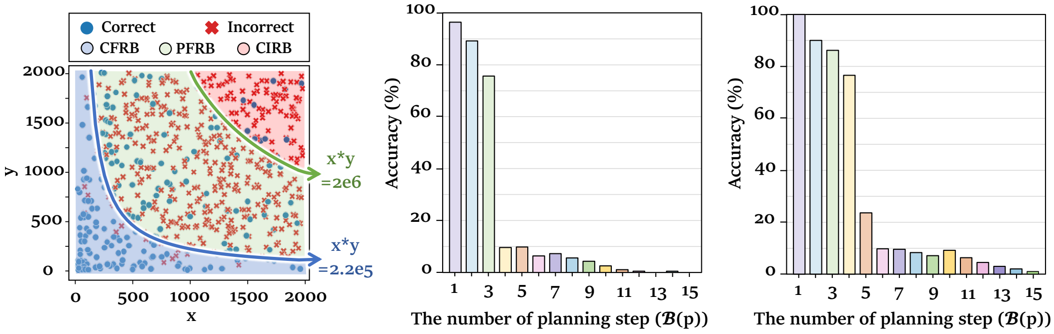
2.3 Categories of Reasoning Boundary

Furthermore, in order to guide the optimization of CoT and more convenient expression, as shown in Figure 1 (c), we define the following three categories of RBs based on their empirical accuracy:

Completely Feasible Reasoning Boundary: We define that the part with an accuracy greater than 90% is a completely feasible reasoning boundary ($\texttt{CFRB} = \mathcal{B}_{\text{Acc}\geq 90\%}(t_1, t_2, \ldots, t_n|m)$), which means that LLMs can effectively grasp the performance of this part.

Completely Infeasible Reasoning Boundary: We believe that the part with an accuracy less than 10% is a completely infeasible reasoning boundary ($\texttt{CIRB} = \mathcal{B}_{\text{Acc}\leq 10\%}(t_1, t_2, \ldots, t_n|m)$), which means that the model can never effectively grasp the performance of this part.

Partially Feasible Reasoning Boundary: We define the RB in the rest part except CFRB and CIRB as a partially feasible reasoning boundary ($\texttt{PFRB} = \mathcal{B}_{10\%<\text{Acc}<90\%}(t_1, t_2, \ldots, t_n|m)$), which requires the model to repeat thinking or more clear information to solve the problem.



(a) Distribution of correct predictions for x*y samples.

(b) Distribution of correct predictions for nature language planning.

(c) Distribution of correct predictions for code planning.

Figure 2: Existence Verification for Reasoning Boundary. Figures (b, c) present evaluations performed on BigGSM, where the reasoning paths are manually analyzed to identify the specific steps at which errors occur, without considering whether the final conclusions are correct.

We analyze the nature of these three categories of RB in detail (in Section 4.3), and further utilize the combination law to optimize these three reasoning boundaries (in Section 5), so as to provide effective suggestions and guidance to support future CoT optimization.

3 Experimental Setup

Benchmark Settings To assess the reasoning boundaries of LLMs, we require a dataset rich in RB. This necessitates tasks with evenly distributed complexities and reasoning steps that challenge the models' upper-bounds. To meet these requirements, we introduce BIGGSM, a new dataset offering greater calculation complexity and longer reasoning chains. The detailed construction process for BIGGSM is provided in Appendix C.

Model Settings Except for model expansion experiments, all experiments are conducted on GPT-3.5-Turbo. Following the setting of Wei et al. [2022], in our CoT experiment, all multi-step reasoning tasks utilize three manually constructed demonstrations. In addition, for all the experiments, top-p is selected from $\{0.95, 1\}$. Temperature is selected from $[0, 1]$ and serves as the main error variable.

4 Empirical Analysis of Reasoning Boundary

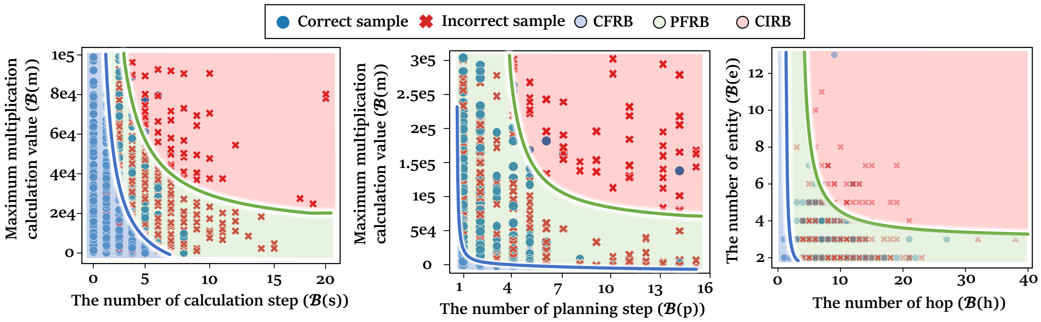
4.1 Existence Verification for Reasoning Boundary

In this study, we investigate the hypothesis that an LLM exhibits varying levels of reasoning boundary across various tasks. To this end, we will verify whether the model has widespread reasoning boundary in various tasks in the following three tasks:

Basic Arithmetic Calculation First, to investigate the existence of RB, we first examine basic arithmetic operations (including addition, subtraction, multiplication, and division). As illustrated in Figure 2 (a), the results reveal significant performance variations across three distinct regions. For multiplication, accuracy surpasses 90% for results up to $2.2e5$. Conversely, accuracy falls below 10% for products exceeding $2e6$. Similar presences of varying RBs are observed for other operations, which verifies the existence of reasoning boundary in basic arithmetic calculation tasks. Further results and implementation details are provided in Appendix D.

Nature Language Planning We further investigate RB in natural language planning tasks for mathematical reasoning. We prompt the model to generate plans and assess their accuracy through manual evaluation. There is a strong correlation between the number of reasoning steps and LLMs' performance in Figure 2 (b). When the model meets the question with fewer than 2 reasoning steps, accuracy surpasses 90%. Conversely, when reasoning steps exceed 4, accuracy falls below 10%. This finding suggests that there are also three different RB categories in natural language planning tasks.

Code Planning For further extensive exploration, we further prompt LLMs by PAL [Gao et al., 2023] to generate code-format plans and evaluate them by manual annotation. As shown in Figure 2 (c), the code planning task is similar to natural language planning, which is also an obvious division



(a) The combination law of different reasoning boundaries in complex calculation task.

(b) The combination law of different reasoning boundaries in mathematical reasoning task.

(c) The combination law of different reasoning boundaries in multi-hop question-answering task.

Figure 3: Combination law verification of RB on different tasks. More verification results on other tasks are shown in Figure 12.

and different categories of RBs. Notably, since code planning utilizes code for clearer logic and reduced expression complexity, its planning accuracy surpasses that of natural language planning.

4.2 Combination Law Verification on Different Tasks

Combination Law in Complex Arithmetic Calculation Building on the proof of Equation (13), we hypothesize that the combination law for RB in the complex arithmetic calculation is the harmonic average of the arithmetic calculation RB and calculation planning RB. To verify this, we designed an experiment focusing on formulas containing addition, subtraction, and multiplication, like "$(1 + 2) * 3 - 4$". Since addition and subtraction complexities are assumed to be around $1e15$ (as shown in Figure 13), the arithmetic calculation RB primarily depends on the multiplication RB and calculation planning RB. Therefore, as shown in Figure 3 (a), there are two obvious RB lines, namely $\mathcal{B}_{Acc=90\%}$ and $\mathcal{B}_{Acc=10\%}$, which are completely consistent with the combination law of these basic RB based on the Equation (2). Besides, these two lines also clearly divide the RBs into three categories.

Combination Law in Mathematical Reasoning Inspired by Tan [2023b], Xiao and Liu [2024], we posit that the natural language mathematical CoT task is determined by two sub-tasks: step planning task and step calculation task for global logic planning and local mathematical calculation. Furthermore, each model output step requires a single basic operation, resulting in a step calculation boundary close to the maximum number of multiplications, denoted by $\mathcal{B}(c) \approx \mathcal{B}(m)$. Formally, with step planning RB denoted by ($\mathcal{B}(p)$) and the step calculation RB by ($\mathcal{B}(c)$), then the combined RB satisfies the following law:

$$\mathcal{B}^{\text{CoT}}(c,p) = \frac{1}{\frac{N_1}{(\mathcal{B}(c)-b_1)} + \frac{N_2}{(\mathcal{B}(p)-b_2)}}. \tag{3}$$

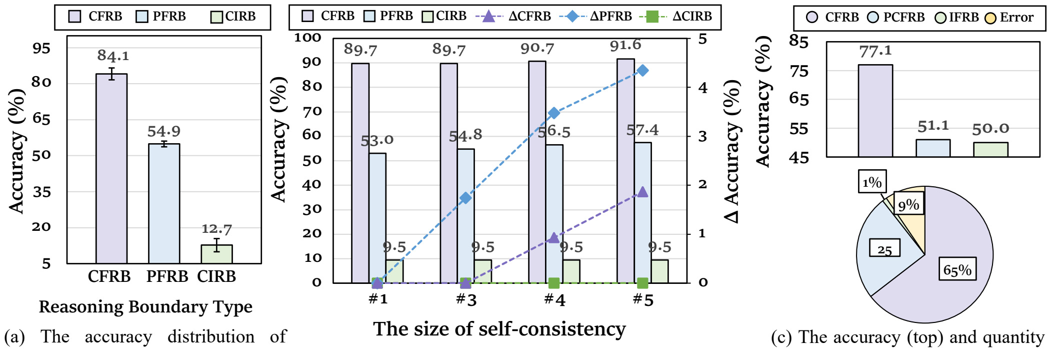
As illustrated in Figure 3 (b), the actual performance distribution of RB (including $\mathcal{B}_{Acc=90\%}$ and $\mathcal{B}_{Acc=10\%}$) in natural language mathematical reasoning task fully aligns with the proposed combination law in Equation (3). Additionally, there are also obviously three RBs in Figure 3 (b).

Combination Law in Multi-hop Reasoning Beyond the realm of mathematics, we further extend our exploration of the combination law to the field of multi-hop question answering. Specifically, we validate our law on HotpotQA [Yang et al., 2018], where we define the reasoning boundary as the combination of global hop-planning RB and local knowledge entity reasoning RB. As shown in Figure 3 (c), $\mathcal{B}_{Acc=90\%}$ and $\mathcal{B}_{Acc=10\%}$ also satisfy the weighted harmonic mean of these two sub-reasoning boundaries. It is also proved that, in addition to math-related tasks, multi-hop question answering also satisfies our proposed combined law and also exhibits three distinct RBs. We will describe in detail how to calculate the combination law on multi-hop reasoning in Appendix E.

4.3 Nature Analysis for different Reasoning Boundary

According to the definition of different RBs, we have divided the problem into three parts for LLMs. In this section, we will verify whether the defined RB adheres to the intrinsic nature of the model itself. We will discuss the natures of these RBs in detail:

`CFRB` **means complete mastery of the model even without demonstration.** According to the definition, we assume that a question within `CFRB` implies a comprehensive understanding of the



(a) The accuracy distribution of generated rationales based on Auto-CoT and Zero-CoT.

(b) Model self-consistency integrated performance in different Reasoning Boundary areas.

(c) The accuracy (top) and quantity (bottom) distribution of synthetic samples from Synthetic-CoT.

Figure 4: Nature analysis at different reasoning boundaries. For Figure (c), all demosntrations in CIRB are special value points obtained by calculation methods similar to 25000×1000. In fact, no real CIRB demosntrations are obtained.

associated issue for a certain LLM. To verify this, following Zhang et al. [2022] and Wei et al. [2022], we formulate a mathematical request and generate chain-of-thought rationale and answer through zero-shot prompting without any demonstration. As shown in Figure 4 (a), it still achieves 29.2% improvement in CFRB on generating the correct rationale compared to other RBs. This also proves that the model can indeed master tasks well on the questions in CFRB.

PFRB **means moderate confidence in its solution and needs consensus building process.** To gauge the level of performance and confidence, we draw parallels to human decision-making, where moderate confidence often necessitates multiple times of consensus building. Inspired by this, we investigate it on Self-Consistency [Wang et al., 2022], which integrates results from various reasoning answers to reach a conclusive answer. Figure 4 (b) demonstrates that as the integration of reasoning paths increases, the accuracy improves significantly within PFRB compared with other RBs. This suggests that within PFRB, the LLM exhibits moderate confidence in solving problems, which needs multiple consensus building.

CIRB **exhibits poor reasoning performance even with consensus building.** As illustrated in Figure 4 (a), questions in CIRB display extremely low accuracy (around 9.5%). And the model shows consistently poor performance and no improvement on Self-consistency in this boundary in Figure 4. It signifies that the model exhibits poor reasoning performance.

LLM has self-awareness of its own RBs. In parallel, a natural question arises: *Is the model capable of discerning its inherent RBs?* To investigate this, we employ the Synthetic-CoT [Shao et al., 2023] to prompt LLM to generate CoT data. As depicted in Figure 4 (c), the results demonstrated that there are over 65% of generated samples within CFRB, which achieves a much higher percentage and performance than other RBs. This suggests that LLMs possess an intrinsic understanding of their RBs and constraints to generate the task they grasp, indicative of a potential for self-assessment.

Takeaways: (1) Reasoning boundary (RB) and the combination law of RB are both widespread across a series of tasks. (2) Different categories of RB can reflect the corresponding performance, and the model can also have a self-understanding of its own RB.

5 RB-based CoT Optimization

5.1 How can we improve CoT by optimizing RB?

Based on our framework, the reasoning boundary limits the performance of the model. The simplest approach to improve CoT is to optimize the step calculation RB $\mathcal{B}(c)$ to promote the value of RB. Specifically, Tool-Usage [Paranjape et al., 2023] and Program-of-Thought (PoT) [Chen et al., 2024b] have shown significant success in CoT optimization. We explain the rationale behind their effectiveness, why PoT consistently outperforms direct Tool Usage [Yao et al., 2023, Chen et al., 2023], and take them as examples to demonstrate how to improve CoT by promoting RB.

Model	BIGGSM		
	Acc. (↑)	Input Token (↓)	Output Token (↓)
`CoT`	57.00 $_{\pm 0.93}$	780.43	96.76 $_{\pm 3.22}$
	RB-Optimized Methods		
`Tool Usage`	71.64 $_{\pm 0.66}$	688.43	129.53 $_{\pm 3.82}$
`PoT`	78.25 $_{\pm 1.09}$	657.43	78.25 $_{\pm 1.09}$
	Reasoning-Path-Optimized Methods		
`Least-to-most`	58.25 $_{\pm 3.28}$	679.59	176.09 $_{\pm 15.22}$
`Complex-CoT`	59.78 $_{\pm 0.60}$	1111.43	131.82 $_{\pm 1.91}$
`CoT+MARP`	64.37 $_{\pm 2.24}$	614.43	95.12 $_{\pm 0.77}$
`PoT+MARP`	**80.55** $_{\pm 2.40}$	**576.43**	**76.34** $_{\pm 2.84}$

Table 1: Main experimental results on GPT-3.5-Turbo. Results on different benchmarks are shown in Table 2.

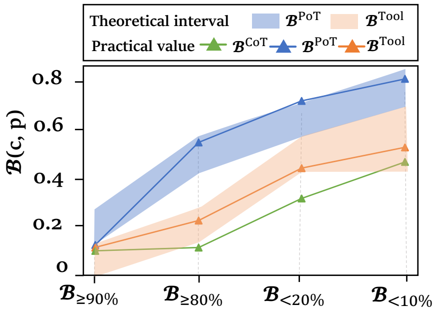


Figure 5: Analysis of the impact of Tool-Usage and PoT on reasoning boundary $\mathcal{B}(c, p)$.

Tool Usage can boost the value of RB for an LLM. When the model uses tools [Paranjape et al., 2023], we can simply think that the model can perform calculations with infinite precision, so that the RB of mathematical calculations tends to infinity, viz $\mathcal{B}(c) \to +\infty$. It is obvious that the combined RB of the model can be calculated as:

$$\mathcal{B}^{\texttt{Tool}}(c, p) = \lim_{\mathcal{B}(c)\to+\infty} \frac{1}{\frac{N_1}{(\mathcal{B}(c)-b_1)} + \frac{N_2}{(\mathcal{B}(p)-b_2)}} = \frac{\mathcal{B}(p) - b_2}{N_2}. \tag{4}$$

Easy to get, $\mathcal{B}^{\texttt{Tool}}(c, p) > \mathcal{B}^{\texttt{CoT}}(c, p)$, this shows that Tool Usage can improve the boundary of reasoning. This explains why Tool Usage can have better performance than vanilla CoT (as shown in Table 1). Furthermore, as shown in Figure 5, the distribution of theoretical RB and the actual one almost perfectly coincide. This also demonstrates the reliability and applicability of our theory.

Program-of-Thought can further enhance the value of LLM's RB. Equation (4) reveals that an LLM's RB hinges entirely on its planning capability. Since natural language can be verbose, it hinders the planning capability of LLM [Gao et al., 2023, Hu et al., 2023, Puerto et al., 2024, Chen et al., 2024b]. `PoT` [Chen et al., 2023] offers a clearer representation of logic using code, allowing for clearer planning (as shown in Figure 2 (b, c)). This leads to finer-grained planning reasoning $\mathcal{B}^*(p) > \mathcal{B}(p)$. Then the PoT reasoning boundary $\mathcal{B}^{\texttt{PoT}}(c, p) > \mathcal{B}^{\texttt{Tool}}(c, p)$, aligning with the observed performance gains of `PoT` over `Tool Usage` (see Table 1). Furthermore, Figure 5 visually demonstrates that PoT's theoretical and practical reasoning boundaries consistently outperform `Tool Usage`. This reinforces the theoretical advantage of PoT and its empirical effectiveness.

5.2 How can we improve CoT based on a certain RB?

Enhancing RB is crucial for optimizing CoT, but requires changes to the model or its reasoning architecture to be effective. Therefore, we need to consider how to optimize the reasoning path so that the difficulty satisfies the RB ($d^* = \mathcal{B}_{Acc=K_1}$) instead of the original RB ($d = \mathcal{B}_{Acc=K_2}$), where $K_2 < K_1$. According to Equation (3), $\mathcal{B}$ is affected by both arithmetical RB and planning RB. Given $\mathcal{B}$, we consider optimizing reasoning ability from the following two strategies as examples [2]:

Complex CoT (`CCoT`): By increasing the boundary of planning to reduce the pressure of single-step calculation, reduce the arithmetical RB, and then get smaller d; However, it introduces more planning steps, which adds the planning pressure. As shown in Figure 6, the model performance first increases and then decreases with the increasing number of `CCoT` steps.

Least-to-Most (`LtM`): By dividing multiple sub-questions to reduce the pressure of local planning within a sub-question, reduce the boundary of local planning, and then get smaller d. However, even though it can release local planning pressure (as demonstrated in Figure 7), this approach simultaneously intensifies global planning pressure by generating an excessive number of sub-questions (as depicted in Figure 15).

Limitation: (1) `CCoT` needs to keep balance in the number of reasoning steps and calculation pressure. (2) Although the pressure of local planning has been reduced, `LtM` has not effectively reduced the pressure of global planning, nor the pressure of optimization calculations.

[2] See detailed analysis for these two strategies in Appendix F.

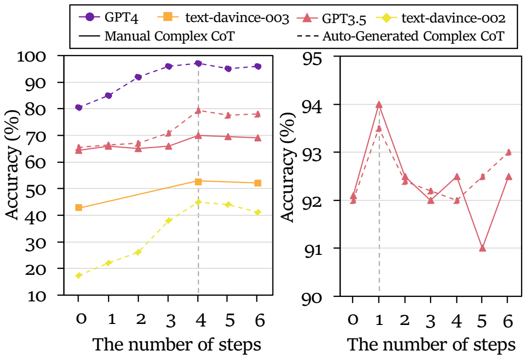


Figure 6: Correlation between the number of steps and performance of Complex-CoT on GSM8K (left) and SingleEq (right). See Appendix G for more meta-analysis results.

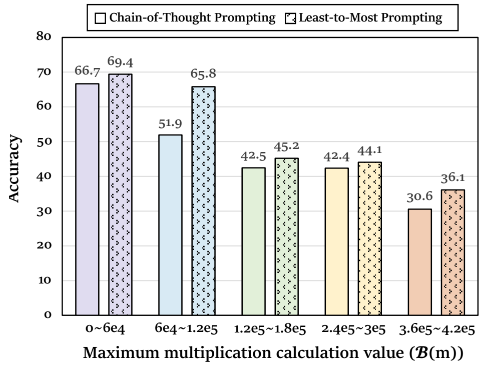


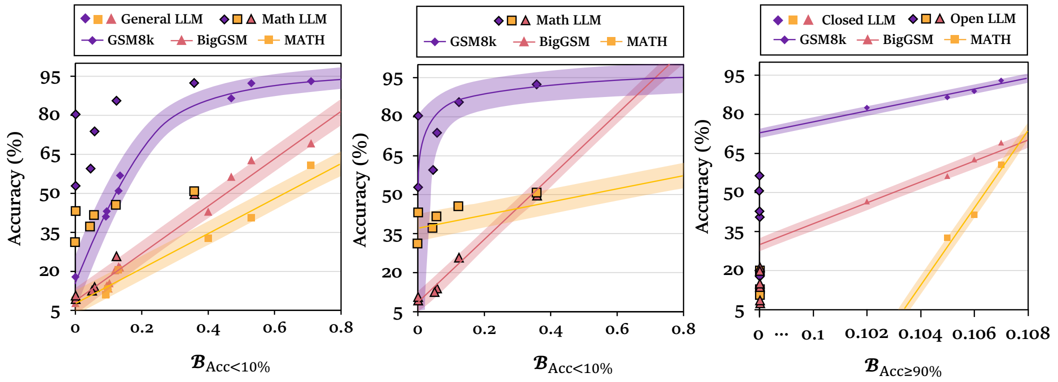
Figure 7: The performance distribution of Least-to-Most prompting on different calculation amounts.

Minimum acceptable reasoning paths prompting can further achieve better CoT within a specific RB. To address the aforementioned two issues, we proposed Minimum Acceptable Reasoning Paths (MARP). Our first objective is to alleviate the computational burden of the model. We achieve this by introducing instructions that set an upper limit on its single-step computational capacity, thereby optimizing the boundary of its computational reasoning. Secondly, we aim to enhance the model's acceptability. Within the calculation and planning boundary, we increase the amount of computation performed in each step in demonstrations as much as possible while simultaneously reducing the number of global planning steps, which effectively mitigates planning pressure. As shown in Table 1, MARP demonstrably improves model performance and effectively reduces the token consumption. By maximizing operations per step, MARP leads to a more streamlined and efficient problem-solving process. Detailed descriptions of this strategy are shown in Appendix G.3.

Takeaways: (1) Tool-Usage and PoT can be utilized to optimize CoT by the calculation and planning reasoning boundary optimization. (2) MARP can well lessen planning and calculation pressure by problem optimization in certain RB (3) Users can effectively optimize CoT performance by optimizing the reasoning boundary and the problem.

6 Expansion Verification & Exploration

RB can be extended to various models. To extend our mechanism's applicability, we verify the mechanism on 25 diverse models (details in Table 3). As shown in Figure 8 (a), we observe a positive correlation between reasoning boundary and model accuracy on mathematical benchmarks.



(a) Correlation between the values of CIRB $\mathcal{B}_{\text{Acc}<10\%}$ for different **general LLMs** and performance on real benchmarks.

(b) Correlation between the values of CIRB $\mathcal{B}_{\text{Acc}<10\%}$ for different **math LLMs** and performance on real benchmarks.

(b) Correlation between the values of CFRB $\mathcal{B}_{\text{Acc}\geq 90\%}$ for different **closed and open LLMs** and performance on real benchmarks.

Figure 8: Correlation between the values of RB for different models and performance on real benchmarks. See Appendix H for more empirical details.

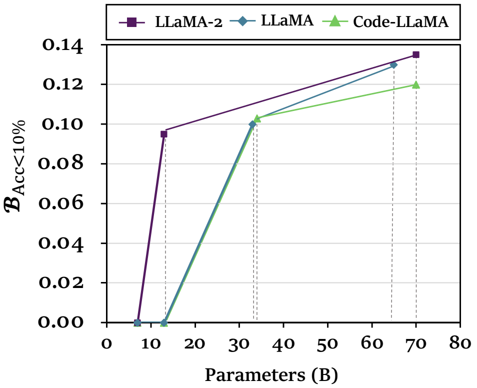


Figure 9: Scaling law correlation between model parameters and CIRB.

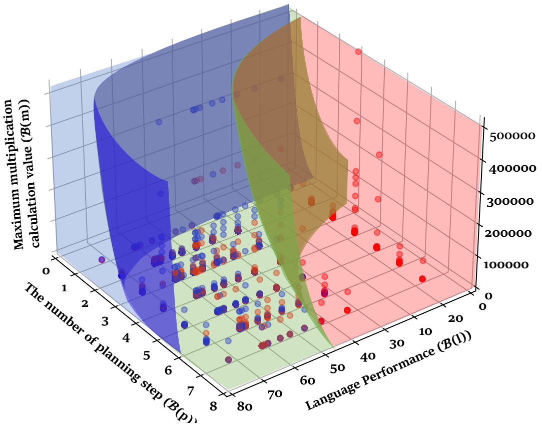


Figure 10: Different boundaries on MGSM.

Moreover, the models that use mathematical data such as MathInstruct for SFT, often have interesting outliers that are different from the general LLMs' area, but they also satisfy a positive correlation with our RBs (as shown in Figure 8 (b)), which helps determine if the model underwent mathematically targeted training.

However, as shown in Figure 8 (c), we find some interesting phenomena. For example, the main difference between the current open-source model and the closed-source model is still CFRB. Except for the closed source model, the CFRB of all models is 0. It shows the potential and the direction of the model optimization. Furthermore, a scaling law of RB can also emerge (as shown in Figure 9): reasoning boundary increased with model parameter count and data quality.

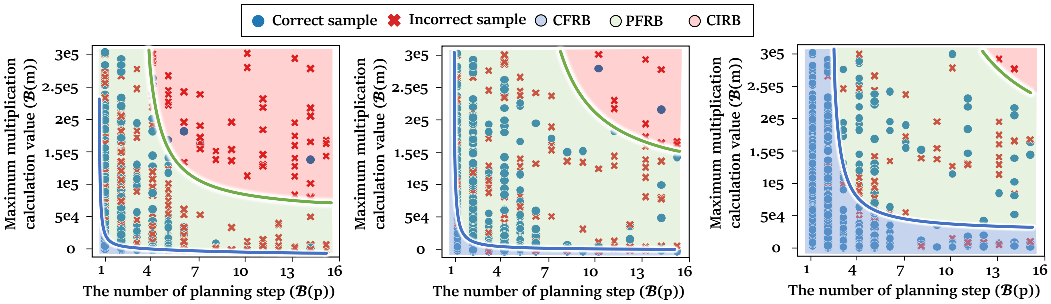
RB can be extended to more tasks. To assess the RB in more tasks, we evaluate them on a multilingual mathematical reasoning task. Inspired by Qin et al. [2024b], we hypothesize that multilingual RB, assessed through direct answer accuracy across different languages, mathematical computation RB, represented by the maximum product result, and reasoning planning RB, indicated by the planning steps, are orthogonal dimensions of performance. We propose that these RBs can be effectively combined using a weighted harmonic mean. As illustrated in Figure 10 confirms that the combined RB maintains the expected three different RBs. Detailed implementation description is shown in Appendix I.

7 Related Work

In this section, we review recent literature related to Chain-of-Thought (CoT) prompting, focusing on theoretical and empirical investigations. Madaan et al. [2023], Wang et al. [2023a], Saparov and He [2023], He-Yueya et al. [2023], Zhang et al. [2024], Wang et al. [2024] and Prystawski et al. [2024] qualitatively show that the LLMs learn the reasoning chain based on the demonstrations in the context. Besides, Lampinen et al. [2022] and Tan [2023a] find a causal link between generated intermediate steps and the final answers during a series of qualitative experiments. Wang et al. [2023c], Hanna et al. [2024] and Dutta et al. [2024] study neural substructure within the LLMs, embodying CoT reasoning from a white-box mechanism perspective, demonstrating that LLMs deploy multiple parallel answer generation paths internally.

Recently, a large amount of work has demonstrated the upper-bounds and limitations of LLM in various CoT tasks [Qin et al., 2023, Imani et al., 2023, Huang et al., 2024, Sprague et al., 2024]. Bi et al. [2024] investigate these bounds on planning capability in code generation by training LLM on CoT samples of varying difficulties. Their findings suggest that LLMs have a limited capacity to learn or manage tasks exceeding a certain complexity threshold. Further understanding of the CoT upper-bound, Merrill and Sabharwal [2023], Li et al. [2023] and Feng et al. [2024] analyze single-step arithmetic capability, which suggests an upper bound on model performance related to input length in single-step reasoning processes.

Despite advancements in CoT explanation for LLMs, significant challenges remain, including the absence of quantifiable metrics for CoT's upper-bounds and the deficiency in optimization guidelines.



(a) GPT-3.5-turbo (b) GPT-4o (c) O1-preview

Figure 11: Combination law verification of reasoning boundaries on GPT-series models.

To tackle this, we propose a reasoning boundaries framework (RBF) to systematically quantify and optimize various CoT approaches. This framework offers a transferable and user-friendly methodology to enhance model performance from a mechanistic perspective. We anticipate that it will furnish systematic insights for ongoing research and inform future developments in the field.

8 Discussion

Discussion on the Boundaries Improvements Furthermore, in order to better understand the best existing LLMs, we utilize RBF to test the current most advanced GPT-series models. As shown in Figure 11, all reasoning boundaries improve a lot compared to the last version which also achieves performance enhancement. Notably, the CFRB increases slightly compared with the improvement of CIRB between GPT-3.5 and GPT-4o. But o1 significantly improves the CFRB. Furthermore, as shown in Figure 14 in Appendix, o1 shows extremely significant improvements on CFRB, which is almost three times of other models. We attribute it to the fact that the advanced Reinforce-Learning and Inference Scaling strategies play a key role in improving this part of the ability compared with the normal improvements in CFRB, which might trigger more in-depth research.

Broader impacts. Our framework is the first work to quantify the reasoning upper-bound of LLMs. This enables the explanation for a huge part of the valid CoT framework. We hope that our work can provide new insights and more systematic guidance for future interpretability analysis of CoT. For social impact, this work may have a certain impact on the controllable and explainable AGI.

Limitations & Future. Due to the cost and time constraints, this work does not discuss the complex relationships such as causal conditions among the basic RBs. In addition, evaluating the robustness and applicability of CoT reasoning boundaries-related techniques in dynamic scenarios will be crucial for future research.

9 Conclusion

This study introduces a novel reasoning boundaries framework (RBF) to quantify and optimize the limitations of LLMs in CoT tasks. Specifically, we propose the concept of reasoning boundaries (RBs) and the combination law of RBs in more complex scenarios for quantitative metrics. We further introduce three categories of RB for CoT optimizations. The framework is validated through extensive experiments across 27 models and 5 tasks. Furthermore, we improve the CoT in both RB and question optimization perspectives to achieve state-of-the-art performance in BIGGSM. We hope that this framework paves the way for further research on understanding and enhancing LLMs' reasoning capabilities.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62236004, 62441603, 62476073 and 62306342. This work was also sponsored by the Excellent Young Scientists Fund in Hunan Province (2024JJ4070), the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3024, and the CCF-Zhipu.AI Large Model Innovation Fund (NO.CCF-Zhipu202406).

References

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

Zhen Bi, Ningyu Zhang, Yinuo Jiang, Shumin Deng, Guozhou Zheng, and Huajun Chen. When do program-of-thought works for reasoning? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17691–17699, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M^{3}CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.446. URL https://aclanthology.org/2024.acl-long.446.

Weize Chen, Chenfei Yuan, Jiarui Yuan, Yusheng Su, Chen Qian, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Beyond natural language: Llms leveraging alternative formats for enhanced reasoning and communication. *arXiv preprint arXiv:2402.18439*, 2024b.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

Daixuan Cheng, Shaohan Huang, and Furu Wei. Adapting large language models via reading comprehension. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y886UXPEZ0.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Subhabrata Dutta, Joykirat Singh, Soumen Chakrabarti, and Tanmoy Chakraborty. How to think step-by-step: A mechanistic understanding of chain-of-thought reasoning. *arXiv preprint arXiv:2402.18312*, 2024.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2024.

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=yf1icZHC-l9.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10764–10799. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/gao23f.html.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did Aristotle Use a Laptop? A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 04 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00370. URL https://doi.org/10.1162/tacl_a_00370.

Michael Hanna, Ollie Liu, and Alexandre Variengien. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36, 2024.

Joy He-Yueya, Gabriel Poesia, Rose Wang, and Noah Goodman. Solving math word problems by combining language models with symbolic solvers. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*, 2023.

Mengkang Hu, Yao Mu, Xinmiao Chelsey Yu, Mingyu Ding, Shiguang Wu, Wenqi Shao, Qiguang Chen, Bin Wang, Yu Qiao, and Ping Luo. Tree-planner: Efficient close-loop task planning with large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Yi Hu, Haotong Yang, Zhouchen Lin, and Muhan Zhang. Code prompting: a neural symbolic method for complex reasoning in large language models. *arXiv preprint arXiv:2305.18507*, 2023.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12365–12394, 2023.

Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R Lyu. How far are we on the decision-making of llms? evaluating llms' gaming ability in multi-agent environments. *arXiv preprint arXiv:2403.11807*, 2024.

Shima Imani, Liang Du, and Harsh Shrivastava. Mathprompter: Mathematical reasoning using large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 37–42, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

Song Jiang, Zahra Shakeri, Aaron Chan, Maziar Sanjabi, Hamed Firooz, Yinglong Xia, Bugra Akyildiz, Yizhou Sun, Jinchao Li, Qifan Wang, et al. Resprompt: Residual connection prompting advances multi-step reasoning in large language models. *arXiv preprint arXiv:2310.04743*, 2023b.

Mingyu Jin, Qinkai Yu, Haiyan Zhao, Wenyue Hua, Yanda Meng, Yongfeng Zhang, Mengnan Du, et al. The impact of reasoning step length on large language models. *arXiv preprint arXiv:2401.04925*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. Can language models learn from explanations in context? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.38. URL https://aclanthology.org/2022.findings-emnlp.38.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2023.

Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. What makes chain-of-thought prompting effective? a counterfactual study. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1448–1535, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.101. URL https://aclanthology.org/2023.findings-emnlp.101.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2023.

Meta. Llama 3. 2024.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. In *Deep Learning for Code Workshop*, 2022.

OpenAI. Introducing chatgpt. 2022.

OpenAI. Gpt-4 technical report, 2023.

Wenbo Pan, Qiguang Chen, Xiao Xu, Wanxiang Che, and Libo Qin. A preliminary evaluation of chatgpt for zero-shot dialogue understanding. *arXiv preprint arXiv:2304.04256*, 2023.

Bhargavi Paranjape, Scott Lundberg, Sameer Singh, Hannaneh Hajishirzi, Luke Zettlemoyer, and Marco Tulio Ribeiro. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.

Ben Prystawski, Michael Li, and Noah Goodman. Why think step by step? reasoning emerges from the locality of experience. *Advances in Neural Information Processing Systems*, 36, 2024.

Haritz Puerto, Martin Tutek, Somak Aditya, Xiaodan Zhu, and Iryna Gurevych. Code prompting elicits conditional reasoning abilities in text+ code llms. *arXiv preprint arXiv:2401.10065*, 2024.

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

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. Large language models meet nlp: A survey. *arXiv preprint arXiv:2405.12819*, 2024a.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*, 2024b.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*, 2023.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=qFVVBzXxR2V.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. In *International Conference on Machine Learning*, pages 30706–30775. PMLR, 2023.

Kashun Shum, Shizhe Diao, and Tong Zhang. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, 2023.

Zayne Rea Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jenyYQzue1.

Jiashuo Sun, Yi Luo, Yeyun Gong, Chen Lin, Yelong Shen, Jian Guo, and Nan Duan. Enhancing chain-of-thoughts prompting with iterative bootstrapping in large language models. *arXiv preprint arXiv:2304.11657*, 2023.

Juanhe (TJ) Tan. Causal abstraction for chain-of-thought reasoning in arithmetic word problems. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 155–168, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.12. URL https://aclanthology.org/2023.blackboxnlp-1.12.

Juanhe TJ Tan. Causal abstraction for chain-of-thought reasoning in arithmetic word problems. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 155–168, 2023b.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *arXiv preprint arXiv:2402.10176*, 2024.

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

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

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. On the planning abilities of large language models-a critical investigation. *Advances in Neural Information Processing Systems*, 36:75993–76005, 2023.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.153. URL https://aclanthology.org/2023.acl-long.153.

Haoyu Wang, Hongming Zhang, Yueguan Wang, Yuqian Deng, Muhao Chen, and Dan Roth. Are all steps equally important? benchmarking essentiality detection in event processes. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4048–4056, 2023b.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? *arXiv preprint arXiv:2402.18272*, 2024.

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

Yiqun Wang, Sile Hu, Yonggang Zhang, Xiang Tian, Xuesong Liu, Yaowu Chen, Xu Shen, and Jieping Ye. How large language models implement chain-of-thought? 2023c.

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

Changnan Xiao and Bing Liu. A theory for length generalization in learning to reason. *arXiv preprint arXiv:2404.00560*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, 2018.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

Yufeng Zhang, Xuepeng Wang, Lingxiang Wu, and Jinqiao Wang. Pattern-aware chain-of-thought prompting in large language models. *arXiv preprint arXiv:2404.14812*, 2024.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2022.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Ziyu Zhuang, Qiguang Chen, Longxuan Ma, Mingda Li, Yi Han, Yushan Qian, Haopeng Bai, Zixian Feng, Weinan Zhang, and Ting Liu. Through the lens of core competency: Survey on evaluation of large language models. *arXiv preprint arXiv:2308.07902*, 2023.

Appendix

A Mathematical Analysis & Proof

A.1 Definitions & Assumptions

In order to further quantify and analyze the combination law of RB, we will define the concept of difficulties for different tasks:

Definition 1 *The difficulty of solving a certain problem during model reasoning is an independent constant.*

That is, the difficulty $\mathcal{D}(t_1, t_2)$ satisfies:

$$\mathcal{D}(t_1, t_2|m) = \mathcal{D}(t_1|m) + \mathcal{D}(t_2|m) = K_1 + K_2, \tag{5}$$

where, K_1, K_2 denotes the relevant constants. Therefore, the combined difficulty formally satisfies:

$$\mathcal{D}(t_1, t_2, \ldots, t_n|m) = \mathcal{D}(t_1, t_2, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n|m) + \mathcal{D}(t_i|m) = \sum_i \mathcal{D}(t_i|m) \tag{6}$$

Definition 2 *The RB is defined as the reciprocal of the difficulty of solving the problem. The greater the difficulty of solving the problem, the lower the RB and the smaller the feasible area.*

Therefore, the combination law of RB satisfies:

$$\mathcal{B}(t_1, t_2, \ldots, t_n|m) \propto \frac{1}{\mathcal{D}(t_1, t_2, \ldots, t_n|m)} \tag{7}$$

Definition 3 *If all basic RBs are infinite, it means that all the difficulties approach to the zero and the model is omnipotent. Therefore, the combined RB is also infinite.*

Formally, the combination law satisfies that:

$$\mathcal{B}(+\infty, +\infty, \ldots, +\infty|m) = +\infty \tag{8}$$

Assumption 4 *The combination law function is continuously differentiable everywhere.*

Assumption 5 *All basic reasoning boundary for combined reasoning boundary are mutually independent.*

A.2 The Proof of Combination Law

Based on the above definitions and assumptions, we need to prove that the combination law is a combined RB and is the weighted harmonic average of two basic RBs.

Proof. Following Equation (7), we can get the $\mathcal{D}(x_1, x_2, \ldots, x_n|m)$ as:

$$\mathcal{D}(x_1, x_2, \ldots, x_n|m) = \sum_{i=1}^{n} \mathcal{D}(0, \ldots, x_i, \ldots, 0|m). \tag{9}$$

According to the Taylor expansion formula, we expand this formula at $x_i \to k_i$, we can get:

$$\begin{aligned}
\mathcal{D}(x_1, x_2, \ldots, x_n|m) &= \sum_{i=1}^{n} \sum_{j=1}^{+\infty} N_{ij}(x_i - k_i)^j && (10)\\
&= \sum_{i=1}^{n} N_{i1}(x_i - k_i) + \mathcal{O}(x_i) && (11)\\
&\approx \sum_{i=1}^{n} N_{i1}(x_i - k_i), && (12)
\end{aligned}$$

where $N_{i1} = \frac{\partial \mathcal{D}(x_1,x_2,\ldots,x_n|m)}{\partial x_i}$. We set $t_i = \frac{1}{x_i} + b_i$ and $\frac{1}{\mathcal{B}(t_1,t_2,\ldots,t_n|m)} \propto \mathcal{D}(x_1,x_2,\ldots,x_n|m)$. Then the original formula is expressed as:

$$\mathcal{B}(t_1,t_2,\ldots,t_n|m) \approx \frac{N_0}{\sum_{i=1}^{n} \frac{N_{i1}}{t_i-b_i} - k_i} + k_0 = \frac{1}{\sum_{i=1}^{n} \frac{N'_{i1}}{t_i-b_i} - k'_i} + k_0, \tag{13}$$

where t_i represents the specific task measurement value, and N_0 and k_0 denote the linear parameters. Given the minimal change in the derivative within the observable range, $N'_{i1} = \frac{N_{i1}}{N_0}$ is treated as a constant N_i in this task for simplicity. Experimental results show that, if sub-RBs are separated independently, $k'_i = \frac{k_i}{N_0}$ and k_0 is typically 0. Since t_i cannot be directly quantified, we use basic form of $\mathcal{B}(t_i|m)$ as its quantized substitute, thus simplifying the combination law as:

$$\mathcal{B}(t_1,t_2,\ldots,t_n|m) \approx \frac{1}{\sum_{i=1}^{n} \frac{N_i}{\mathcal{B}(t_i|m)-b_i}}. \tag{14}$$

A.3 Calculation of RB in Practical Process

To determine the constants, we first fit parameters to a model using a development dataset (or 20% of the test dataset if the development dataset is not available). This fitting process yields the corresponding constants. For a given task and prompt strategy, these constants remain fixed. Additionally, once the combination law constants are established, different reasoning boundaries are determined through a binary search on performance in a standard setting (3-shot CoT). For instance, we use binary search to identify a reasoning boundary that ensures the accuracy of all problems below that boundary approaches 90%, achieving $\mathcal{B}_{Acc=90\%}$. For one model, one task, and one prompt type, the reasoning boundary remains fixed. Zero-shot and few-shot settings share the same set of reasoning boundaries.

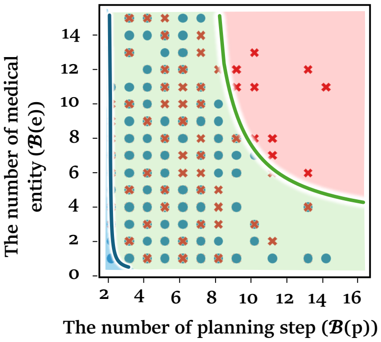


Figure 12: Extended verification of combination law on Medical Knowledge Probing [Cheng et al., 2024] tasks.

B The Application Tutorial of Reasoning Boundary

From a practical standpoint, our mechanism framework exhibits universal adaptability, making it suitable for application in a wide range of scenarios. When confronted with a new problem context, the framework enables a systematic approach to problem-solving. A key feature of the framework is its reliance on the weighted harmonic mean, which imparts advantageous mathematical properties to its structure. Specifically, the framework operates effectively if the reasoning process can be segmented into relatively independent boundaries. This segmentation allows the framework to be fully leveraged in addressing diverse problems.

Reasoning Boundary Application. In the case of a vertical domain problem based on CoT reasoning, the process can be divided into two key boundary levels: task planning and domain-specific reasoning. These can be modeled as follows:

$$\mathcal{B} = \frac{1}{\frac{1}{\mathcal{B}_p} + \frac{1}{\mathcal{B}_v} + k_1}, \tag{15}$$

where: $\mathcal{B}_p$ represents the task planning boundary, $\mathcal{B}_v$ represents the vertical domain boundary, and k_1 is a constant reflecting the degree of boundary independence.

Reasoning Boundary Definition & Segmentation. Neglecting any of these boundaries only results in an increase in k, but keeping the overall efficiency of the framework. If the reasoning boundary is well-defined and independent, the value of k approaches zero, showcasing the effectiveness of our mechanism framework.

Further Reasoning Boundary Segmentation. Further refinement of the vertical domain boundary, $\mathcal{B}_v$, into $\mathcal{B}_{v1}$ and $\mathcal{B}_{v2}$ is straightforward. No additional complexity is introduced, as the following relationship holds:

$$\mathcal{B}_v = \frac{1}{\frac{1}{\mathcal{B}_{v1}} + \frac{1}{\mathcal{B}_{v2}} + k_2}. \tag{16}$$

Thus, the overall boundary equation can be extended to:

$$\mathcal{B} = \frac{1}{\frac{1}{\mathcal{B}_p} + \frac{1}{\mathcal{B}_{v1}} + \frac{1}{\mathcal{B}_{v2}} + k_1 + k_2}. \tag{17}$$

This formulation allows for flexible and systematic boundary division at multiple levels, enhancing the framework's practical utility across various problem domains.

Challenging Reasoning Boundary Measurement. In addition, we propose an alternative method to measure the reasoning boundaries. This approach allows the model to provide direct answers without relying on CoT reasoning steps. By doing so, the model's reasoning process for a specific task depends solely on a single reasoning boundary, which can be represented as follows:

$$\mathcal{B} = \frac{1}{\frac{1}{\mathcal{B}_1} + k_1}. \tag{18}$$

For instance, in the MGSM task, assessing multilingual reasoning boundary is particularly challenging. To address this, we directly evaluate the model's performance using a direct prompting strategy without CoT outputs and use this performance to define the multilingual reasoning boundary, which in turn helps determine the corresponding normalization constant. Subsequently, we apply multilingual CoT reasoning to the MGSM task to calculate the combined boundary using the framework's combination law. This approach provides a more generalized solution and may be more adaptable to specific needs.

C Details of Dataset

C.1 Dataset Construction

To adequately assess the reasoning boundary of LLMs, it is essential to develop a dataset that encompasses a range of complexities and reasoning boundaries. To address these challenges, we propose a novel approach to constructing a mathematical reasoning dataset using manual synthesis and annotation which finally leads to the BigGSM benchmark. Specifically, our proposed method involves the manual synthesis and annotation of a mathematical reasoning dataset. The construction process includes the following steps:

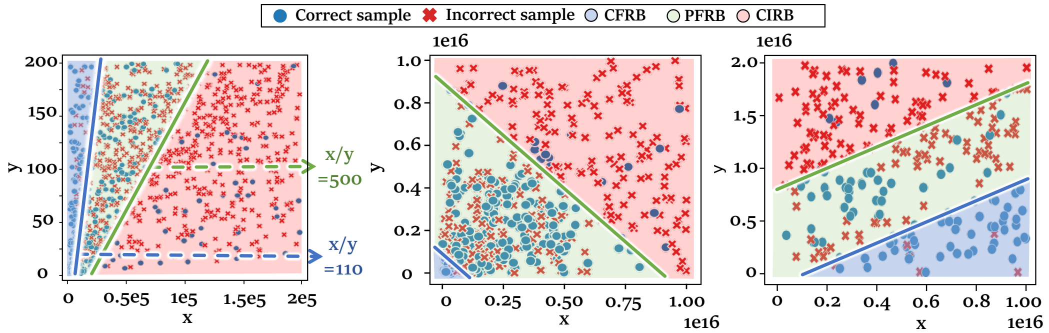
Step 1: Domain Template Generation Initially, we employ a prompt-driven LLM (GPT-4) to generate complex scenarios necessitating multi-step calculations. This process also yields initial example templates. Specifically, the prompt given to the large model is as follows:

> Generate a scenario-related template involving multiple mathematical steps to solve a real-world problem. Ensure the scenario requires the application of different mathematical concepts. Please use "[VAR]" as a variable to mark the template of the question.

Step 2: Natural Language Template Creation Recognizing that LLMs can produce errors and logical inconsistencies, we refine these initial templates to improve their accuracy and add mathematical calculations. To facilitate the generation of extended sequences, we decompose the templates into smaller, loopable segments that incrementally meet the multi-step reasoning demands.

Step 3: Domain Template Augmentation To address the limited diversity in individual samples and provide a broader evaluation of LLMs' mathematical abilities, we use an LLM (GPT-4) to generate at least three alternative augmented templates for each original template and step. The generation prompt we use is as follows:

> Create three alternative versions of the following template that introduce different complexities or variables, ensuring each version demands an equivalent level of reasoning.

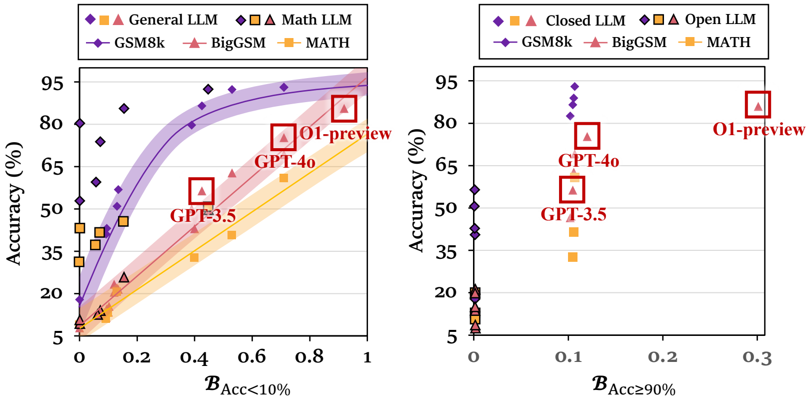


(a) Distribution of correct predictions for x/y samples.

(b) Distribution of correct predictions for x+y samples.

(c) Distribution of correct predictions for x-y samples.

Figure 13: Existence verification for reasoning boundaries on basic arithmetic calculation tasks, including division, addition, and subtraction operations.



(a) Correlation between the values of CIRB $\mathcal{B}_{Acc<10\%}$ for different **general LLMs** and performance on real benchmarks.

(b) Correlation between the values of CFRB $\mathcal{B}_{Acc\geq 90\%}$ for different **closed and open LLMs** and performance on real benchmarks.

Figure 14: Correlation between the values of RB for different models and performance on real benchmarks.

Step 4: Numeric Filling Once all templates are prepared, we aim to test the upper-bound of the LLMs' computational reasoning boundary by introducing numerical values ranging from 1 to 1e5 in multiplication tasks. This step is designed to thoroughly assess the models' performance across a spectrum of numerical challenges.

Step 5: Manual Annotation To ensure the quality and logical coherence of our synthetic samples, we manually review them to correct any errors introduced during the automated generation process. Finally, we hired three experts to mark whether the samples in the data set were correct. Only for those samples where more than two experts agreed did we retain the corresponding samples. The Cohen's kappa value marked by the experts was 0.97, which indicates the perfect agreement.

C.2 Dataset Analysis

Our dataset comprises 610 test samples, which is extensive when compared to the GSM8K dataset. It features a broader range of procedural steps, varying from 1 to 16 steps. Additionally, our dataset encompasses a wider spectrum of computational efforts, ranging from 6 to 3e5.

D The Implementation Details of Basic Arithmetic Calculation

D.1 Data Construction

This section outlines the process used to construct datasets for examining the existence of reasoning boundaries (RB) in basic arithmetic calculations. Initially, we identify the operations for investigation, namely addition, subtraction, multiplication, and division. We then determine the range of integer operands (x and y), starting from 1 to $1e10$, subsequently extending to $1e20$. A random number generator is employed to produce independent and unbiased pairs of x and y within the specified range. For each pair, we compute the expected correct outcome of the chosen operation using standard arithmetic procedures. In addition, in order to ensure that decimals do not affect the computational complexity, we restrict our analysis to integer operands and outcomes to control for complexity and randomly generate numerical values of x and y.

D.2 Prompt Construction

The prompt configuration in our study involves inputting the structured data into a computational model to analyze the arithmetic accuracy. The following prompting is used for LLMs' input:

> Please calculate the formula given below:
> x **op** y=

where **op** denotes the arithmetic operation (selected from addition, subtraction, multiplication, division). And x and y values are generated from Section D.1. The final experimental results are shown in Figure 13.

E The Implementation Details of Multi-hop Reasoning

We propose that the natural language multi-hop CoT task comprises two sub-tasks: multi-hop planning and knowledge step reasoning for multi-hop question answering. To address the challenge of measuring knowledge difficulty, we utilize a NER model[3] to identify the number of knowledge entities in each hop, thus marking the knowledge step reasoning RB in the single-step task. Formally, let $\mathcal{B}(h)$ represent the RB of multi-hop planning and $\mathcal{B}(e)$ denote the RB of knowledge step reasoning. The combined RB satisfies the following combination law:

$$\mathcal{B}^{\texttt{CoT}}(e,h) = \frac{1}{\frac{N_1}{(\mathcal{B}(e)-b_1)} + \frac{N_2}{(\mathcal{B}(h)-b_2)}}. \tag{19}$$

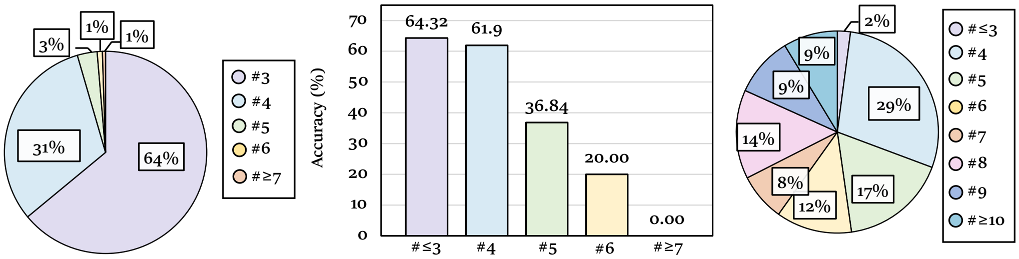
F Analysis for Complex-CoT and Least-to-Most within Reasoning Path Optimization Perspective

Complex CoT Prompting can achieve better CoT within a specific RB by simplifying the calculation reasoning step. We believe that Complex CoT optimizes the performance of the model by allowing the model to reach its computational limit as much as possible in single-step reasoning. Therefore, the combined RB for Complex-CoT can be expressed as:

$$\mathcal{B}^{\texttt{Complex}}(p,c) = \lim_{\mathcal{B}(c)\to\mathcal{B}_{\text{Acc}=100\%}(c)} \frac{1}{\frac{N_1}{(\mathcal{B}(c)-b_1)} + \frac{N_2}{(\mathcal{B}'(p)-b_2)}} \tag{20}$$

Assuming the premises of RB remain unchanged ($\mathcal{B}^{\texttt{Complex}}(p,c) = \mathcal{B}^{\texttt{CoT}}(p,c)$), it can obviously yield the solution $\mathcal{B}'(p) > \mathcal{B}(p)$. Therefore, the model can accept more steps of reasoning boundary, that is, if the planning difficulty d_p is less than reasoning capability $\mathcal{B}'(p)$, the accuracy is higher. In order to analyze this problem, we adopted a meta-analysis method. We count the performance of the work of Jin et al. [2024], Fu et al. [2023] using Complex CoT. The relationship between the performance label and the number of steps is shown in Figure 6 (left). For most multi-step reasoning tasks, generally speaking, within a certain range, as the number of steps increases, the computational pressure of the model is relieved and the performance is improved, which is consistent with the theory and exploration of Feng et al. [2024], Wang et al. [2023b], Valmeekam et al. [2023].

[3] https://huggingface.co/dslim/bert-base-NER



(a) Distribution of **the number of CoT steps** performed for each sub-question.

(b) The **Accuracy Distribution** on the CoT steps performed for each sub-question.

(c) Distribution of the **number of sub-question**.

Figure 15: Analysis of output results for Least-to-Most Prompting.

However, We can also clearly recognize the flaws of Complex CoT. Once the difficulty of planning d_p (that is, the number of planning steps) is greater than $\mathcal{B}'(p)$, it exceeds the capabilities of the model and the performance will decline. We can observe that for single-step calculation reasoning, as shown in Figure 6 (right), the performance of using Complex CoT will gradually decrease. The rest of mathematical reasoning will also decrease when the number of steps is greater than a certain threshold. This phenomenon can also be explained by our combination law. While the amount of calculation is reduced, the number of reasoning steps is also increasing. If the acceptable number of reasoning steps is exceeded, the reasoning boundary is exceeded, and the model performance will decline, which demonstrates that it is necessary to keep a balance between the number of reasoning steps and computational pressure (see Appendix G for detailed meta-analysis process).

Limitation: Need to keep balance in the number of reasoning steps and calculation pressure.

Least-to-Most Prompting can achieve better CoT within a specific RB by simplifying the planning reasoning paths. Least-to-most prompting structures problem-solving hierarchically, by breaking questions into smaller sub-questions and further solving them one-by-one. Accordingly, the Least-to-most RB can be divided into three sub-RBs, namely, the problem decomposition RB $\mathcal{B}(d)$, the problem planning RB $\mathcal{B}(p)$, and the single-step calculation RB $\mathcal{B}(c)$. Therefore, the combined RB for least-to-most can be expressed as:

$$\mathcal{B}^{\texttt{LtM}}(d,p,c) = \frac{1}{\frac{N_1}{(\mathcal{B}'(c)-b_1)} + \frac{N_2}{(\mathcal{B}(p)-b_2)} + \frac{N_3}{(\mathcal{B}(d)-b_3)}}. \tag{21}$$

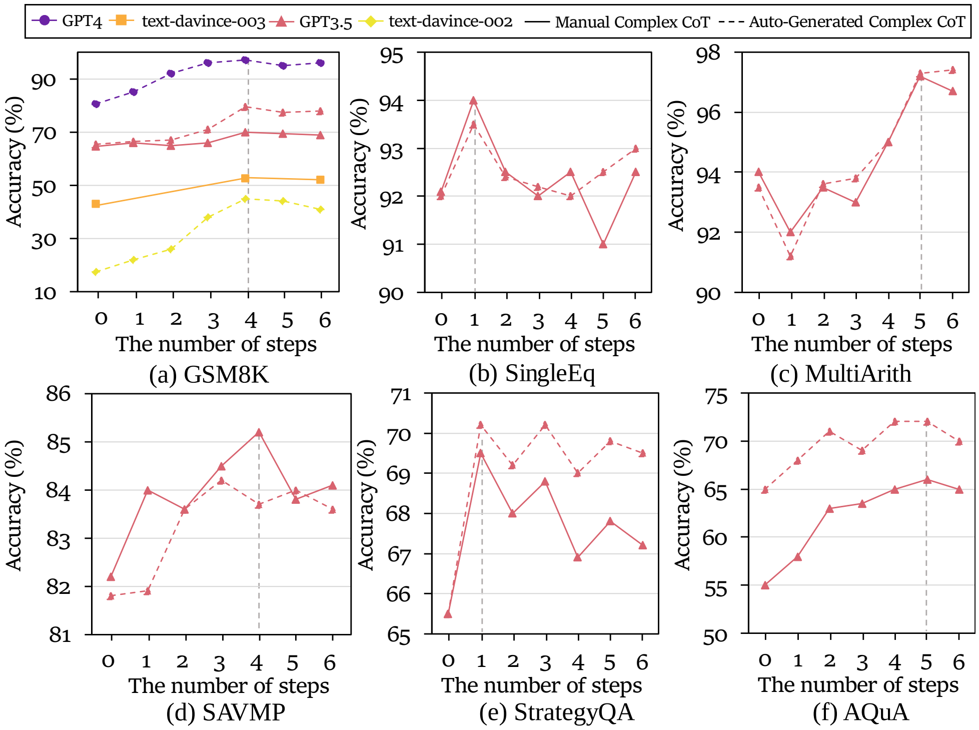
Ideally, if the problem decomposition ability of the model is excellent ($\mathcal{B}(d) \to +\infty$), it can decompose the problem into sub-problems that can be solved in one step every time $\mathcal{B}(c) \to 1$, therefore the least-to-most RB can be expressed as:

$$\hat{\mathcal{B}}^{\texttt{LtM}}(d,p,c) = \lim_{\mathcal{B}(c)\to 1, \mathcal{B}(d)\to +\infty} \mathcal{B}^{\texttt{LtM}}(d,p,c) = \frac{\mathcal{B}'(c)-b_2}{N_1(\mathcal{B}'(c)-b_2)-N_2}, \tag{22}$$

Assuming the premises of RB remain unchanged ($\hat{\mathcal{B}}^{\texttt{LtM}}(d,p,c) = \mathcal{B}^{\texttt{CoT}}(p,c)$), it can obviously yield the solution $\mathcal{B}'(c) > \mathcal{B}(c)$. On the contrary, the model can accept larger difficulty d, which also shows that using least-to-most prompting can effectively increase the maximum of acceptable calculation RB under a given RB (as shown in Figure 7), thereby improving model performance. As shown in Table 1, we find that LLM can be optimized by Least-to-most from vanilla CoT.

However, the performance improvement of the model is not significant, which we attribute to the fact that the current model cannot push its performance to the ideal limit. As shown in Figure 15 (a), the reasoning boundary of the model cannot make each reasoning step completely tend to 1, which also leads to the difference in reasoning performance in Figure 15 (b). In the meantime, the model's ability to divide problems is also limited. What's more, as shown in Figure 15 (c), in around 90% of cases, the model will only divide less than 6 problems, which also limits the performance.

Limitation: Although the pressure of local planning has been reduced, it has not actually effectively reduced the pressure of global planning, nor the pressure of optimization calculations.



(a) GSM8K (b) SingleEq (c) MultiArith

(d) SAVMP (e) StrategyQA (f) AQuA

Figure 16: The effectiveness on average step length in demonstrations for CoT performance.

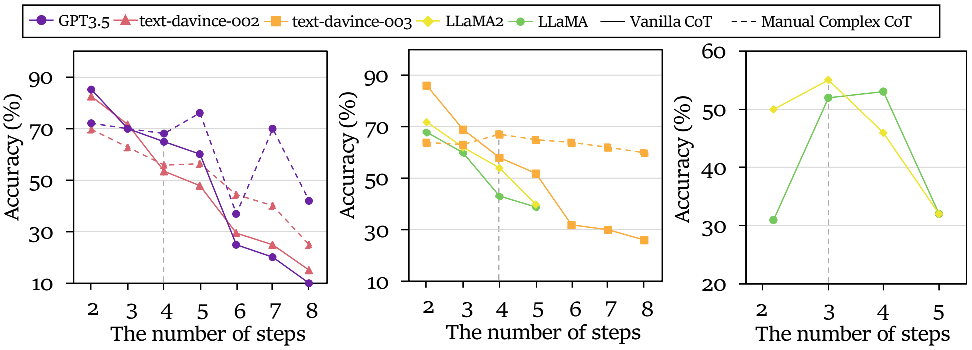
G The Meta-Analysis for Complex-CoT Prompting

In order to discuss the advantages and limitations of Complex-CoT, we conduct two detailed meta-analyses through two distinct perspectives. The first one assesses the influence of the reasoning steps demonstrated in In-Context Learning (ICL) across various tasks, while the second evaluates the effects of employing a fixed number of reasoning steps in ICL on questions with different reasoning steps. These meta-analyses aim to compare the efficacy of these methods against prior studies systematically. Specifically, we conducted a systematic search for relevant studies addressing the same problem tackled by Jin et al. [2024], Fu et al. [2023], Shum et al. [2023], Sun et al. [2023], and Jiang et al. [2023b]. We ensured that retrieved studies were pertinent, focusing on studies that addressed the same problem and used similar evaluation metrics.

G.1 The effectiveness of step length in demonstrations

From each selected study, including Jin et al. [2024] and Fu et al. [2023], we evaluate the performance using Complex CoT. The relationship between performance and the number of Complex CoT's steps is shown in Figure 16. For most multi-step reasoning tasks, as the number of steps increases within a certain range, the computational load decreases, and performance improves.

However, as described in Appendix F, the flaws of Complex CoT are apparent. When the difficulty of planning (d_p), defined as the number of planning steps, exceeds $\mathcal{B}'(p)$, the model's capabilities are surpassed, leading to a performance decline. This is evident in single-step calculation reasoning, as shown in Figure 16 (c, e), where performance using Complex CoT gradually decreases. Similarly, for other mathematical reasoning tasks, performance decreases when the number of steps exceeds a certain threshold. This phenomenon aligns with our combination law: while reducing the amount of calculation, the number of reasoning steps increases. Exceeding the acceptable number of reasoning steps surpasses the reasoning boundary, causing a decline in model performance. Therefore, maintaining a balance between the number of reasoning steps and computational pressure is crucial.



(a) GSM8K (b) GSM8K (c) AQuA

Figure 17: The effectiveness of step length in golden samples with a fixed step length in demonstrations for CoT performance.

G.2 The effectiveness on step length in golden samples

Furthermore, to gain a nuanced understanding of the impact of Complex CoT when the number of steps exceeds the golden step number, we conduct further meta-analysis from Fu et al. [2023], Shum et al. [2023], Sun et al. [2023], Jiang et al. [2023b]. Specifically, as illustrated in Figure 17 (a, b), our analysis reveals that for problems of low complexity and with smaller golden step numbers, Complex CoT tends to underperform compared to Vanilla CoT. Notably, it is only when the reasoning steps exceed two that Complex CoT outperforms Vanilla CoT. This suggests that Complex CoT effectively optimizes single-step computations and enhances model performance for complex problems. However, it increases the cognitive load for simple problems, resulting in a performance decline.

Interestingly, this phenomenon is also observed with the simpler Vanilla CoT, as shown in Figure 17 (c). The model achieves significant performance gains only when the number of reasoning steps aligns with the target output steps. If the complexity of the planned steps exceeds the necessary reasoning boundary, or if there is no effective optimization for reasoning boundary, the performance deteriorates.

G.3 The Implementation Details of Minimum Acceptable Reasoning Paths

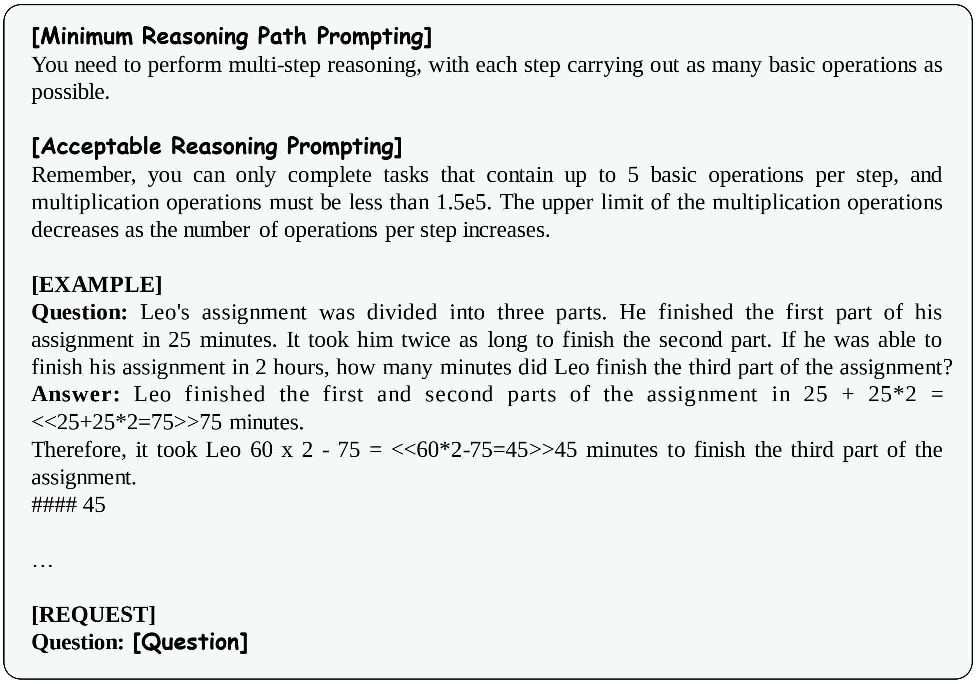
To address the two aforementioned limitations, we propose Minimum Acceptable Reasoning Paths (MARP). Firstly, to reduce the model's computational load, we introduce instructions that limit its single-step computing power, thereby optimizing its reasoning boundary. Secondly, to enhance the model's acceptability, we increase the computation amount per step within this boundary and reduce the number of global planning steps, thus alleviating planning pressure.

To control variables effectively, we make only the simplest modifications to the prompt to achieve the desired CoT optimization.

Minimum Reasoning Path Prompting To alleviate the cognitive load associated with planning, it is essential to have the model respond to the question as succinctly as possible. This approach ensures that the focus remains on providing a short, clear and direct reasoning path. The following prompt is designed to achieve this objective:

> You need to perform multi-step reasoning, with each step carrying out as many basic operations as possible.

Acceptable Reasoning Prompting To effectively utilize the model, it is crucial to define the upper-bound of reasoning boundary. This ensures that the complexity of the reasoning process is manageable and within acceptable bounds. The specific prompt to achieve this is as follows:



[Minimum Reasoning Path Prompting]
You need to perform multi-step reasoning, with each step carrying out as many basic operations as possible.

[Acceptable Reasoning Prompting]
Remember, you can only complete tasks that contain up to 5 basic operations per step, and multiplication operations must be less than 1.5e5. The upper limit of the multiplication operations decreases as the number of operations per step increases.

[EXAMPLE]
Question: Leo's assignment was divided into three parts. He finished the first part of his assignment in 25 minutes. It took him twice as long to finish the second part. If he was able to finish his assignment in 2 hours, how many minutes did Leo finish the third part of the assignment?
Answer: Leo finished the first and second parts of the assignment in 25 + 25*2 = <<25+25*2=75>>75 minutes.
Therefore, it took Leo 60 x 2 - 75 = <<60*2-75=45>>45 minutes to finish the third part of the assignment.
45

…

[REQUEST]
Question: [Question]

Figure 18: Minimum acceptable reasoning path prompting for natural language chain-of-thought. All examples given in the context transform from Wei et al. [2022].

Model	Acc. (↑)	Input Token (↓)	Output Token (↓)
HotpotQA [Yang et al., 2018]			
CoT	289.50	67.27	26.50
CoT+MARP	309.51	68.39	28.73
Medical Probing [Cheng et al., 2024]			
CoT	636.11	249.78	48.9
CoT-MRP	476.11	86.52	69.41
StrategyQA [Geva et al., 2021]			
CoT	1046.28	225.35	63.90
CoT+MARP	649.28	167.40	74.09

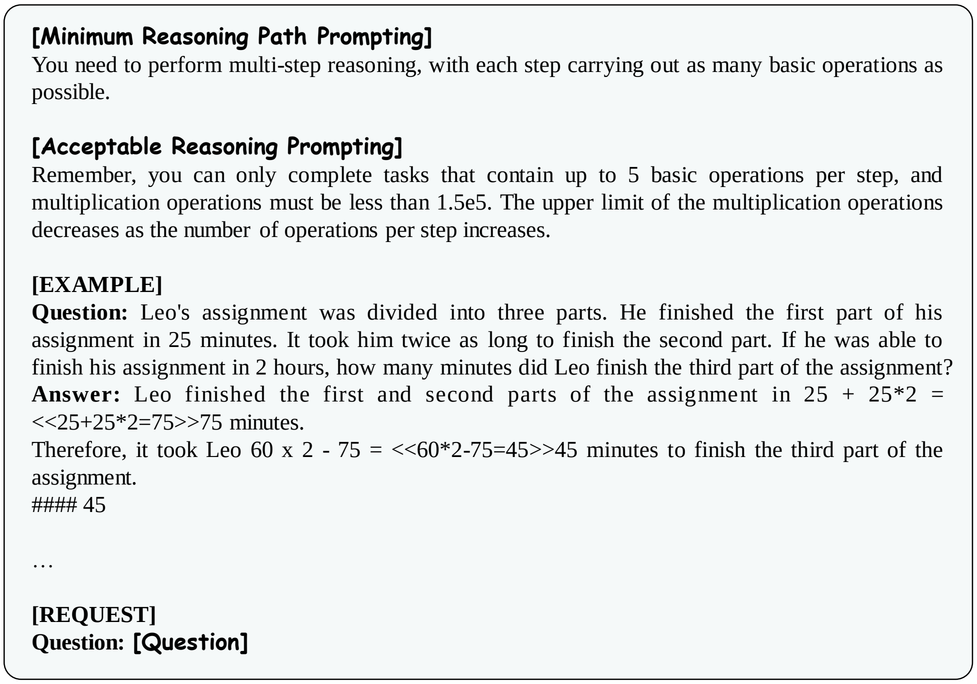
Table 2: Extended experimental results on GPT-3.5-Turbo.

> Remember, you can only complete tasks that contain up to 5 basic operations per step, and multiplication operations must be less than 1.5e5. The upper limit of the multiplication operations decreases as the number of operations per step increases.

This prompt is designed to set clear boundaries for the model's operations, thereby optimizing its performance and accuracy.

Furthermore, it is necessary to enhance the demonstration within the corresponding in-context learning framework to meet the specific needs of our Model-Agnostic Reasoning Protocol (MARP). This involves refining the examples and instructions provided to ensure they align perfectly with the MARP requirements. Figures 18 and Figure 19 illustrate our MARP prompt, showcasing how to structure the demonstrations to facilitate effective learning and reasoning in natural language CoT and program-of-thought setting. By adhering to these guidelines, we can ensure that the model operates efficiently and produces reliable results.

In summary, setting precise boundaries for reasoning boundary and optimizing in-context learning demonstrations are essential steps in enhancing the model's performance. By following the specified



[Minimum Reasoning Path Prompting]
You need to perform multi-step reasoning, with each step carrying out as many basic operations as possible.

[Acceptable Reasoning Prompting]
Remember, you can only complete tasks that contain up to 5 basic operations per step, and multiplication operations must be less than 1.5e5. The upper limit of the multiplication operations decreases as the number of operations per step increases.

[EXAMPLE]
Question: Leo's assignment was divided into three parts. He finished the first part of his assignment in 25 minutes. It took him twice as long to finish the second part. If he was able to finish his assignment in 2 hours, how many minutes did Leo finish the third part of the assignment?
Answer: Leo finished the first and second parts of the assignment in 25 + 25*2 = <<25+25*2=75>>75 minutes.
Therefore, it took Leo 60 x 2 - 75 = <<60*2-75=45>>45 minutes to finish the third part of the assignment.
45

…

[REQUEST]
Question: [Question]

Figure 19: Minimum acceptable reasoning path prompting for program-of-thought. All examples given in the context transform from Wei et al. [2022].

prompt and refining the MARP examples, we can achieve a high level of accuracy and efficiency in the model's reasoning processes.

H The Implementation Details in various LLMs

We employ 25 commonly used models to evaluate the extensibility of our framework to a broader range of models. The specific models are listed in Table 3. For each model, we utilize the chat/instruct version whenever available to maximize their ability to follow instructions. Additionally, we deploy all models on the vLLM [Kwon et al., 2023] framework to ensure a fair comparison. Except for model OpenMath-series [Toshniwal et al., 2024] which does not conform to the vLLM format, all other models are deployed on vLLM for testing. All experiments on open-source models were conducted on two A100 80G. Following the setting of Wei et al. [2022], in our CoT experiment, all multi-step reasoning tasks are with three manually constructed demonstrations. In addition, for all the experiments, our top-p is selected from $\{0.95, 1\}$, and temperature is selected from $[0, 1]$.

In addition, the only difference in the prompt is that we use different dialogue delimiters to make it conform to the format of the LLM instruction fine-tuning, thereby avoiding the bias caused by the gap between training and inference.

I The Implementation of Combination Law in MGSM

Inspired by Qin et al. [2023] and Huang et al. [2023], we propose that the multilingual mathematical CoT task comprises three sub-tasks: step planning, step calculation, and multi-modal expression. We evaluate the model's mathematical expression ability in different languages based on its zero-shot direct performance on MGSM, as reported by Qin et al. [2023]. For relevant parameter calculations, please see the "Challenging Reasoning Boundary Measurement" part of Appendix B. Formally, let step planning RB be denoted by $\mathcal{B}(p)$, step calculation RB by $\mathcal{B}(c)$, and multilingual expression RB by $\mathcal{B}(l)$. The combined RB satisfies the following law:

$$\mathcal{B}^{\texttt{CoT}}(c,p,l) = \frac{1}{\frac{N_1}{(\mathcal{B}(c)-b_1)} + \frac{N_2}{(\mathcal{B}(p)-b_2)} + \frac{N_3}{(\mathcal{B}(l)-b_3)}}. \tag{23}$$

Model	Base Model	Parameters (B)
Open-source General LLM		
LLaMA [Touvron et al., 2023a]	-	7, 13, 33, 65
LLaMA-2 [Touvron et al., 2023b]	-	7, 13, 70
LLaMA-3 [Meta, 2024]	-	8, 70
Code-LLaMA [Roziere et al., 2023]	LLaMA-2 [Touvron et al., 2023b]	7, 13, 34, 70
Mistral [Jiang et al., 2023a]	-	7
Close-source General LLM		
Gemini-1.0-Pro [Team et al., 2023]	-	-
GPT3.5-Turbo [OpenAI, 2022]	-	-
Claude-3-Haiku [Anthropic, 2024]	-	-
Claude-3-Sonnet [Anthropic, 2024]	-	-
Claude-3-Opus [Anthropic, 2024]	-	-
GPT4 [OpenAI, 2023]	-	-
Open-source Math LLM		
MAmmoTH [Yue et al., 2023]	LLaMA-2 [Touvron et al., 2023b]	7,13
MAmmoTH [Yue et al., 2023]	Mistral [Jiang et al., 2023a]	7
OpenMATH-Instruct [Toshniwal et al., 2024]	LLaMA-2 [Touvron et al., 2023b]	70
OpenMATH-Instruct [Toshniwal et al., 2024]	Mistral [Jiang et al., 2023a]	7

Table 3: Model list. In order to ensure a certain ability to follow instructions, we use the Instruct version of the model as much as possible (if available).

As shown in Figure 10, the performance distribution of RB (including $\mathcal{B}_{Acc=90\%}$ and $\mathcal{B}_{Acc=10\%}$) in the multilingual mathematical reasoning task aligns with the proposed combination law in Equation (23). Moreover, three distinct RBs are evident in Figure 10.

J Ethical Considerations

Data Access. Our data is adapted from GSM8K [Cobbe et al., 2021] and supplemented with manually created samples. GSM8K is an open-source dataset available for academic research.

Dataset Collection Process. We began with an introductory task interview using 50 example questions, compensating participants $20 each to familiarize themselves with the task. During the annotation process, annotators were paid $15 per hour, totaling approximately 60 hours of work.

The Rest of Data Annotation Process. For the remaining data annotation, we hired a graduate student with CET-6 proficiency in Chinese and English and strong mathematical knowledge. The student was compensated $15 per hour, which is above the local average salary. The instructions for annotation are as follows:

> You need to annotate the generated number of steps, maximum computation amount, correctness of the generation steps, correctness of the calculations, and correctness of the model output:
>
> - **Number of generated steps:** This refers to how many reasoning steps the model generated.
> - **Maximum computation amount:** This indicates the largest product of operations in the model's reasoning steps.
> - **Correctness of generation steps:** This assesses the accuracy of the model's planning. If all steps and operators are planned correctly, and the operand values are logically correct, it is considered correct, regardless of calculation accuracy.
> - **Correctness of calculations:** This considers only whether the calculations are correct, ignoring planning factors.
> - **Correctness of the output:** This checks whether the model's final answer is correct.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As shown in lines 5-18 of the Abstract and 59-69 of the Introduction, we present our main claims and outline the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our work in Section 8.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our work is not strictly a purely theoretical work, we provide more of an empirical formula. In addition, we analyze the source of our empirical formula in Appendix F and provide the corresponding assumption and proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: As shown in Appendix C to Appendix I, we have provided detailed descriptions and analyses of the experimental setups for all our investigations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release our code in the official version of the subsequent paper to provide reproduction and provide more help to the future community.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: As shown in Section C to Section H, we have provided detailed descriptions and analyses of the experimental setups for all our investigations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We report error bars in Table 1 and Figure 4, and explain the error variables in Section 3. However, error bars are not reported for all tasks because it would be too expensive for human annotation and computational resource consumption.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: As shown in Section 3 and Appendix H, we provide detailed model compute resources under different settings.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We are convinced that we comply with NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the broader impacts of our work in Section 8. In addition, since our work is more like providing an empirical formula and has no additional social harmfulness, we do not discuss this part.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: As shown in Section J, We describe and analyze the details and ethical considerations of our crowdsourcing in detail.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: Since our region and institution are not required to provide IRB approval, we do not describe this section. We are convinced that our work complies with the NeurIPS Code of Ethics and the guidelines.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.