

Hint-before-Solving: A Framework to Effectively Utilizing Inherent Knowledge of Large Language Model

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have recently showcased remarkable generalizability in various domains. Despite their extensive knowledge, LLMs still face challenges in efficiently utilizing encoded knowledge to develop accurate and logical reasoning. To mitigate this problem, we introduced the **Hint-before-Solving** framework (HinSo), which guides the model in generating hints (e.g., specific knowledge or key ideas) for solving the problem before the step-by-step solution. Our studies involving 5 LLMs across 7 datasets of mathematical and commonsense reasoning, results indicated that introducing hints before problem-solving can significantly enhance the performance of CoT. To investigate whether LLMs can learn the HinSo pattern and improve their generalization ability, we constructed two large-scale and high-quality training datasets, HST-S and HST-L, containing 7.5k and 75k samples, respectively. The experimental results of supervised fine-tuning (SFT) showed that, under the same settings, the performance of model trained on the HinSo-formatted data improved significantly compared to CoT-formatted data, with a performance increase of 5.1% and 5.6% on the GSM8K, respectively. We make our code and dataset publicly available at <https://github.com/sfhff216/hsp>.

1 Introduction

Benefiting from extensive training corpora and computational resources, Large Language Models (LLMs) have reached state-of-the-art performance in numerous Natural Language Processing (NLP) tasks (Touvron et al., 2023a; OpenAI, 2023; Touvron et al., 2023b; Zhao et al., 2023b; Mistral AI Team, 2023). However, LLMs still face challenges in complex reasoning tasks, such as mathematical reasoning (Lu et al., 2023; Luo et al., 2023a; Imani et al., 2023) and commonsense reasoning (Paran-

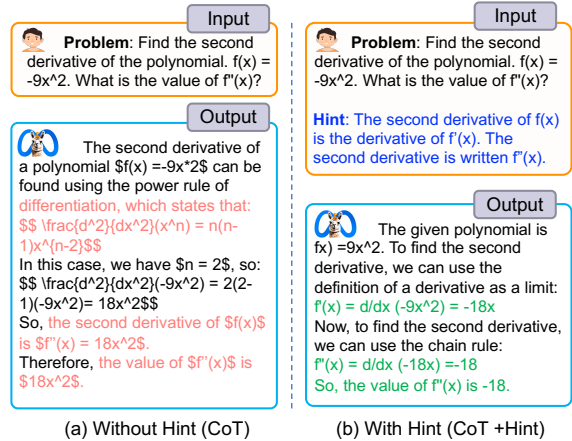


Figure 1: The output comparison of Llama-2-Chat-70B solving a math problem (calculus) with and without a hint. Red text indicates erroneous information; green text indicates correct reasoning. Findings: (1) having a hint can help the LLM understand the problem. (2) The LLM possesses knowledge of calculus, and with a hint, it can accurately apply this knowledge.

jape et al., 2021; Sap et al., 2020). Although possessing a wealth of knowledge, LLMs always fail to accurately apply encoded knowledge to generate coherent and strongly logical reasoning chains when addressing reasoning tasks.

To improve the performance of LLMs on complex reasoning tasks, existing works have made several attempts. These previous works include fine-tuning on complete training datasets (Luo et al., 2023a; Yu et al., 2023; Yue et al., 2023), training-free methods based on prompt engineering (Zhou et al., 2023a; Wang et al., 2023a; Fu et al., 2023; Lyu et al., 2023; Zhao et al., 2023a), or enhancing by retrieving knowledge from external knowledge bases (Yao et al., 2023b; He et al., 2023; Yang et al., 2023). However, supervised fine-tuning methods are resource-intensive, prompt engineering-based methods rarely attempt to improve the ability of LLM to use knowledge accurately, and retrieval-enhancement-based methods are limited to specific tasks. For example, mathematical reasoning that

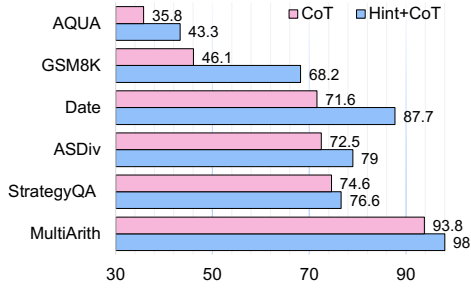


Figure 2: Results for Llama-2-Chat-70B (under CoT prompting) with or without introducing high-quality hints across six reasoning datasets. Findings: introducing hints lead to significant improvements, with an average relative increase of 9.7%.

includes many special symbols is difficult to access relevant knowledge through keyword or semantic retrieval.

To mitigate these problems, in this work, we explore how LLMs can effectively utilize their encoded knowledge to enhance their reasoning logic and performance. We found that providing LLMs with hints effectively guides their use of encoded knowledge for problem-solving. Fig. 1 illustrates this by comparing Llama2-70B’s outputs on a *calculus problem* with and without hints. The LLM cannot utilize *calculus knowledge* to solve the problem without any hints, as shown in Fig.1-(a). However, when given a hint (as shown in Fig.1-(b)): “... *The second derivative is written $f''(x)$.*” the LLM can accurately apply its “*calculus knowledge*” to generate a correct and logical solution with intermediate reasoning. The reason can be attributed to that the hint suggested that “ $f''(x)$ denotes the second derivative”, which helped the LLM to better understand the target of the problem. Moreover, we conducted quantitative analysis on six reasoning datasets by introducing hints generated by GPT-4. The experimental results are shown in Fig. 2. We can find that giving high-quality hints can effectively improve reasoning performance.

However, it is challenging to provide high-quality hints for every sample. To address this problem, we propose the Hint-before-Solving (HS) method, which allows LLMs to generate hints on their own before solving a problem. *The hints may include knowledge necessary for solving the problem (e.g., the hint shown in Fig. 1-(b)), analyzing the question, and providing essential ideas for the solution.* Our explorations of Hint-before-Solving (HinSo) in this paper are driven by following research questions:

Q1: *Can HinSo guiding LLMs to autonomously*

generate helpful hints be effective? To answer this question, we incorporated HinSo into four well-performing prompting methods to investigate how HinSo performs (EXP-I). Furthermore, we examined the effectiveness of the HinSo variant, HinSo2, which provides hints and solutions in two stages (EXP-II). And explore the upper bound of LLMs under the HinSo2 framework (EXP-III). (Sec. 4.1)

Q2: *Does HinSo still work when dealing with tasks that are challenging for LLMs?* In other words, if a task is difficult for LLMs, can they still provide helpful hints? To answer this question, we evaluated the challenging MATH dataset (EXP-IV). Furthermore, we explore how LLMs perform under the self-consistency setting (EXP-V). (Sec. ??)

Q3: *How do LLMs perform if they are supervised fine-tuned on a large-scale HinSo format dataset?* To answer this question, we constructed the HST-S (7.5K) and HST-L (75K) dataset based on GSM8K and conducted supervised fine-tuning on Llemma-7B. The experimental results show that we achieved a performance of **64.3** on Llemma-7B, surpassing GPT3.5. (EXP-VI, Sec. 4.3)

The main contributions of this work are summarized as below:

- (1) We discovered that providing hints allows LLMs to use their encoded knowledge accurately and effectively. For quantitative analysis, with GPT-4 generated hints, Llama-2-Chat-70B’s accuracy increased by nearly 10% across six datasets.
- (2) We propose the HinSo framework, allowing LLMs to automatically generate useful hints. We conducted extensive experiments and analyses on applying HinSo to four popular prompting methods to verify HinSo’s effectiveness.
- (3) We collected two high-quality and large-scale datasets, namely HST-S and HST-L, containing 7.5K and 75K samples enhanced with hints, namely HST-L (to be released), and fine-tuned Llemma-7B to achieve 64.3 accuracy, surpassing GPT-3.5 (57.1) and WizardMath-13B (63.9).

2 Hint before Solving

In this section, we will provide a detailed illustration of the Hint-before-Solving framework. Fig. 3 introduces two categories for incorporating hints: (1) introducing hints from external sources (e.g., knowledge base) to the LLMs (Fig. 3-(a)); (2) introducing hints encoded within the LLM itself, which is further divided into hints provided by training-free LLMs (Fig. 3-(b)), which is strongly relied

Problem: Do black-tailed jackrabbits fear the European wildcat?
Hint: Consider the natural habitats of black-tailed jackrabbits and European wildcats and whether their paths would cross.
Answer: Black-tailed jackrabbits are native to North America. European wildcats are native to Europe. Thus, their paths would not naturally cross. So the answer is no.

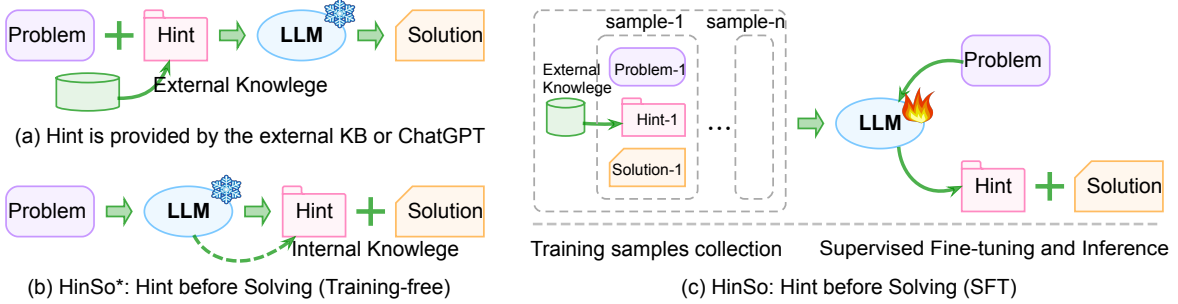


Figure 3: The hint-before-solving framework (HinSo). (a) represents existing research that incorporates external knowledge to the LLM as hints; (b) and (c) illustrate using the LLM’s own knowledge as hints, whereas (b) uses an untrained LLM relying on the prompt design and demonstration samples selection strongly, and (c) requires collecting annotated triplets (problem, hint, solution) for supervised fine-tuning, which will make the LLMs learn the pattern. The purple dashed box illustrates an example of commonsense reasoning with triplet (problem, hint, solution).

on the prompt and demonstrate samples; and hints intuitively introduced by LLMs after learning the hint-before-solving pattern through supervised fine-tuning (Fig. 3-(c)).

What is the hint? *The hints can be the knowledge required for solving the problem, insights for analyzing the question, and key ideas necessary for the solution.* As shown in the box in Fig. 3, there is an example of using the Mixtral-8x7B-Instruct-v0.1 (Mistral AI Team, 2023) to generate the ‘hint’ and ‘answer’ for the given commonsense reasoning problem. For the question, “Do black-tailed jackrabbits fear the European wildcat?”, the LLMs provide the hint “Black-tailed jackrabbits are native to North America. European wildcats are native to Europe. Thus, their paths would not naturally cross. So the answer is no.”, which offers the necessary background knowledge and essential ideas for solving the problem.

External Hints As shown in Fig. 3-(a), many existing studies have explored improving the problem-solving ability of LLMs by retrieving hints from external knowledge bases (Levonian et al., 2023), using stronger language models (e.g., GPT4) (Cohen et al., 2023), or using demonstration samples that are similar to the testing samples (Hu et al., 2022). However, retrieving hint from external sources does not always work. For instance, mathematical problems with complex symbols make it difficult to find reliable and useful knowledge from text or semantics.

HinSo* Therefore, we raise the following re-

search question: Can LLMs provide hints that benefit problem-solving their own to effectively utilize their encoded inherent knowledge while generating chain-of-thought reasoning? Fig. 3-(b) presents the HinSo*, a hint-before-solving framework that is training-free. Given a problem, we design appropriate prompts and demonstration samples to enable the LLM to generate helpful analyses and knowledge hints before solving the problem. In this process, the parameters of the language model are fixed. We leverage the emergent capabilities of LLMs, such as following task instructions (Chung et al., 2022) and in-context learning (Min et al., 2022), to achieve hints before solving. Our HinSo* can be formulized as:

$$P(H, S|T, Q) = P(S|T, Q, H)P(H|T, Q), \quad 198$$

$$P(S|T, Q, H) = \prod_{j=1}^{|S|} \text{LLM}(s_j|T, Q, H, s_{<j}), \quad 199$$

$$P(H|T, Q) = \prod_{i=1}^{|H|} \text{LLM}(h_i|T, Q, h_{<i}), \quad 200$$

where T is the prompt template, Q is the testing problem, H is the hint related to the solving problem, and S denotes the solution. The problem template T for diverse explored tasks can be seen in the appendix D.

HinSo Based on the emergent capabilities of large language models, we have achieved training-free Hint-before-Solving (HinSo*). However, due to the absence of triplets (problem, hint, solution)

in the training data of LLMs, the models cannot intuitively generate hints before producing a solution. HinSo* is fragile and susceptible to the definitions of prompts and the selection of demonstration samples. Therefore, we propose HinSo based on Supervised Fine-Tuning (SFT), as shown in Fig. 3-(c). First, we need to construct training samples for HinSo, which are triplets (problem, hint, solution). Hints can come from manual annotations, retrieved knowledge bases, or be provided by more capable large language models (e.g., GPT4). Here, we collect two high-quality and large-scale training datasets, HinSoTrain-S and HinSoTrain-L, with 7.5k and 75k samples, respectively. Then, conducting supervised fine-tuning based on the constructed training datasets. Finally, perform inference using the LLMs that have been fine-tuned. Our experiments also demonstrate that supervised fine-tuning HinSo requires fewer demonstration samples during the inference stage compared to training-free HinSo.

3 Experiment Setup

| Number | G8K | ASDiv | MArith | AQUA | MATH | SQA | Date |
|----------|-------|-------|--------|------|-------|-------|------|
| Samples | 1,319 | 2,097 | 596 | 254 | 5,000 | 2,290 | 359 |
| Examples | 8 | 8 | 8 | 8 | 4 | 6 | 10 |

Table 1: The number of test samples and prompting examples across seven datasets.

3.1 Large Language Model

To verify the performance of our proposed method, we consider Mixtral-8x7B-Instruct-v0.1 (*Mix-56B*) (Mistral AI Team, 2023) and Llama-2-Chat (Touvron et al., 2023c) family models, where Llama-2-Chat-7B (*Lm2-7B*), Llama-2-Chat-13B (*Lm2-13B*), Llama-2-Chat-70B (*Lm2-70B*) were studied. Note, the italicized text in parentheses represents the abbreviated names of the models.

3.2 Datasets

We evaluated the effectiveness of HinSo framework across multiple datasets for mathematical and common sense reasoning tasks. Tab. 1 shows the number of test samples for these datasets and the number of samples for prompting in a few-shot setting.

Mathematical Reasoning We considered five popular mathematical reasoning datasets, namely *GSM8K* (*G8K*) (Cobbe et al., 2021), *MultiArith* (*MArith*) (Roy and Roth, 2016), *AQuA* (Ling

et al., 2017), *ASDiv* (Miao et al., 2021), and *MATH* (Hendrycks et al., 2021a).

Commonsense Reasoning Two common sense reasoning datasets were also taken into account, which are *StrategyQA* (*SQA*) (Geva et al., 2021) and *Date Understanding* (*Date*) (Srivastava et al., 2022).

3.3 Prompting Methods

The baseline Prompting methods considered in this work are listed below:

(1) *Standard Prompting* (*SD*) (Brown et al., 2020) generates the answer for the given question without intermediate steps. (2) *Chain-of-Thought Prompting* (*CoT*) (Wei et al., 2022) generate step-by-step solutions to a given problem. (3) *Least-to-Most Prompting* (*LtM*) (Zhou et al., 2022) involves decomposing a complex problem into simple subproblems. (4) *Plan-and-Solve Prompting* (*PS*) (Wang et al., 2023b) aims to handle the multi-step reasoning task by planning and solving each plan target.

To validate the effectiveness of the our HinSo framework, we reimplemented some previous prompting methods. *To ensure a fair comparison, we did not deliberately reproduce results reported in previous papers but rather aimed to maintain consistency in the experimental setup. For different prompting methods, we kept using the same set of demonstration samples and modified their format according to the prompting method.* We conducted a performance survey on existing baseline prompting shown in the Appendix F.

3.4 Experimental Settings

Demonstration examples Under any prompting method, one dataset is used with the number of demonstration examples in all the experiments discussed in this work. Specifically, as shown in Tab. 1, there are 8 demonstration examples each of GSM8K, ASDiv, MArith, and AQUA, 6 examples for StrategyQA, 10 examples for Date, 4 examples for MATH.

Hyperparameters of Greedy Decoding We use the vllm library ¹ for few-shot evaluation. For greedy decoding, the hyperparameters are set as: top_p=1, max_tokens=500, temperature=0, and the number of reasoning path n=1. For self-consistency, the number of reasoning path n is set to 4, 16, 32, 64, 128, and temperature = 0.4. Other hyperparameters are set the same as the greedy de-

¹<https://github.com/vllm-project/vllm>

| Method | | HinSo | G8K | ASDiv | MArith | AQUA | SQA | Date | Avg | Improvement | |
|---------|-----|-------|------|-------|--------|------|------|------|------|-------------|--|
| Lm2-7B | SD | × | 5.8 | 43.7 | 7.4 | 19.7 | 62.0 | 33.1 | 28.6 | | |
| | | ✓ † | 5.5 | 44.8 | 6.5 | 21.3 | 63.8 | 39.8 | 30.3 | | |
| | LtM | × | 15.5 | 49.5 | 21.8 | 26.0 | 63.9 | 49.3 | 37.7 | | |
| | | ✓ † | 16.0 | 50.2 | 29.2 | 23.2 | 65.3 | 42.3 | 37.7 | | |
| | PS | × ‡ | 21.8 | 55.8 | 66.6 | 25.6 | 58.1 | 34.8 | 43.8 | | |
| | | ✓ | 21.5 | 56.8 | 60.6 | 25.2 | 60.5 | 33.4 | 43.0 | | |
| | CoT | × | 19.7 | 53.6 | 63.4 | 24.4 | 66.3 | 40.1 | 44.6 | | |
| ✓ † | | 19.9 | 55.8 | 63.8 | 24.4 | 67.5 | 43.2 | 45.8 | | | |
| Rlt Avg | | | 0.0 | 1.2 | 0.2 | -0.4 | 1.7 | 0.3 | 0.5 | | |
| Lm2-13B | SD | × | 8.5 | 48.6 | 10.1 | 19.3 | 65.3 | 40.7 | 32.1 | | |
| | | ✓ † | 8.2 | 49.9 | 11.7 | 21.3 | 68.4 | 55.2 | 35.8 | | |
| | LtM | × | 23.8 | 55.8 | 52.7 | 31.1 | 68.8 | 60.4 | 48.8 | | |
| | | ✓ † | 27.6 | 55.9 | 57.7 | 23.2 | 69.6 | 51.3 | 47.6 | | |
| | PS | × ‡ | 35.1 | 63.0 | 80.7 | 25.6 | 60.9 | 47.6 | 52.2 | | |
| | | ✓ | 32.4 | 62.9 | 74.8 | 25.6 | 66.0 | 50.1 | 52.0 | | |
| | CoT | × | 34.5 | 60.5 | 83.2 | 25.6 | 68.0 | 57.7 | 54.9 | | |
| ✓ † | | 36.5 | 61.2 | 87.1 | 25.6 | 72.1 | 57.7 | 56.7 | | | |
| Rlt Avg | | | 0.7 | 0.5 | 1.1 | -1.5 | 3.3 | 2.0 | 1.0 | | |
| Lm2-70B | SD | × | 12.6 | 60.6 | 26.3 | 24.8 | 72.9 | 54.6 | 42.0 | | |
| | | ✓ † | 12.8 | 62.7 | 25.7 | 25.6 | 75.5 | 76.6 | 46.5 | | |
| | LtM | × | 40.2 | 68.6 | 72.0 | 39.4 | 75.2 | 71.0 | 61.1 | | |
| | | ✓ † | 41.9 | 69.4 | 76.8 | 38.6 | 77.0 | 77.4 | 63.5 | | |
| | PS | × ‡ | 60.0 | 74.1 | 95.8 | 40.2 | 64.7 | 62.4 | 66.2 | | |
| | | ✓ | 55.5 | 72.7 | 93.0 | 36.2 | 58.9 | 63.8 | 63.4 | | |
| | CoT | × | 46.1 | 72.5 | 93.8 | 35.8 | 74.6 | 71.6 | 65.7 | | |
| ✓ † | | 50.3 | 74.4 | 94.6 | 37.0 | 77.0 | 73.0 | 67.7 | | | |
| Rlt Avg | | | 0.4 | 0.9 | 0.6 | -0.7 | 0.2 | 7.8 | 1.5 | | |
| Mix-56B | SD | × | 19.8 | 64.3 | 44.6 | 22.0 | 72.1 | 45.4 | 44.7 | | |
| | | ✓ † | 20.3 | 65.9 | 38.9 | 30.7 | 71.2 | 61.3 | 48.1 | | |
| | LtM | × ‡ | 56.0 | 77.1 | 74.3 | 43.3 | 73.9 | 64.1 | 64.8 | | |
| | | ✓ | 56.0 | 77.0 | 72.8 | 49.2 | 72.4 | 64.3 | 65.3 | | |
| | PS | × ‡ | 73.2 | 84.2 | 97.8 | 49.6 | 66.3 | 68.5 | 73.3 | | |
| | | ✓ | 67.1 | 82.3 | 92.3 | 48.4 | 67.6 | 66.6 | 70.7 | | |
| | CoT | × | 63.7 | 78.3 | 96.1 | 42.5 | 74.7 | 69.9 | 70.9 | | |
| ✓ † | | 69.8 | 80.1 | 97.0 | 48.4 | 75.1 | 77.4 | 74.6 | | | |
| Rlt Avg | | | 0.1 | 0.4 | -2.9 | 4.8 | -0.2 | 5.4 | 1.3 | | |

Table 2: Results of applying HinSo to existing prompting (Sec. 3.3). Green (pink) values indicate the best performance without HinSo (with HinSo). *Rlt Avg* denotes the average relative improvement on the four prompting methods. *Improvement* represents the relative performance improvement when introducing HinSo compared to not using HinSo. † indicates HinSo significantly boosts performance, whereas ‡ suggests omitting HinSo leads to better results.

coding. All inference experiments are based on four A100 GPUs.

| Prompting | Lm2-7B | Lm2-13B | Lm2-70B | Mix-56B |
|-----------|------------|------------|-------------|---------------|
| CoT | 4.5 | 5.6 | 11.1 | 27.0 |
| +HinSo | 4.4 | 5.7 | 11.4 | 28.6 † |

Table 3: Results on MATH dataset. Values in bold denote the best performance, and the value with † denotes the performance of HinSo significantly outperforms CoT.

4 Experiments and Results

4.1 Q1: Can HinSo Work?

To answer this question, we applied HinSo to four existing popular prompting methods to explore how HinSo performs in different prompting meth-

ods. Examples are shown in appendix A. Our experimental prompting methods include standard prompting (SD), Least to Most prompting (LtM), Plan-and-Solve prompting (PS), and CoT prompting, as introduced in Sec. 3.3 The results are shown in Tab. 2. The main findings are summarized as below:

- (1) *HSP is effective in standard and CoT prompting but fails in PS and LtM prompting.* From Tab. 2, we observe that the standard and CoT Prompting show significant performance improvements under HinSo, while the enhancements from PS and LtM are limited. We try to give reasons below: Hints clarify the prompt or problem by offering key insights or solutions, influencing the logic behind the answers. They are crucial in task planning for both PS and LtM prompting, where introducing

| Param. | Prompting | Overall | Type | | | | | | | Level | | | | |
|------------|-----------|---------|-------------|--------------|-------------|-------------|--------------|-------------|--------------|-------------|-------------|--------------|--------------|--------------|
| | | | AG | CP | GT | IA | NT | PG | PC | L1 | L2 | L3 | L4 | L5 |
| n=1,t=0 | CoT | 27.0 | 39.01 | 18.99 | 18.58 | 13.4 | 16.85 | 47.07 | 15.57 | 62.47 | 44.41 | 30.59 | 18.62 | 8.08 |
| | +HinSo | 28.6 | 39.09 | 23.21 | 21.09 | 13.84 | 15.93 | 46.27 | 15.2 | 64.3 | 45.64 | 30.33 | 18.29 | 8.91 |
| | Impv | 1.62 | 0.08 | 4.22 | 2.51 | 0.44 | -0.92 | -0.8 | -0.37 | 1.83 | 1.23 | -0.26 | -0.33 | 0.83 |
| n=4,t=0.4 | CoT | 31.9 | 46.67 | 26.58 | 22.55 | 15.39 | 20.56 | 52.47 | 17.95 | 71.17 | 49.33 | 36.6 | 23.39 | 10.8 |
| | +HinSo | 33 | 47.35 | 26.37 | 26.1 | 15.39 | 21.3 | 54.88 | 19.6 | 72.31 | 51.45 | 36.34 | 25.86 | 11.33 |
| | Impv | 1.1 | 0.68 | -0.21 | 3.55 | 0 | 0.74 | 2.41 | 1.65 | 1.14 | 2.12 | -0.26 | 2.47 | 0.53 |
| n=16,t=0.4 | CoT | 37.6 | 53.41 | 31.22 | 27.35 | 19.38 | 26.67 | 58.9 | 24.73 | 78.03 | 56.71 | 43.15 | 30.07 | 13.52 |
| | +HinSo | 38.8 | 53.75 | 32.07 | 31.52 | 20.93 | 27.59 | 59.82 | 26.01 | 78.49 | 57.83 | 44.39 | 33.11 | 13.44 |
| | Impv | 1.2 | 0.34 | 0.85 | 4.17 | 1.55 | 0.92 | 0.92 | 1.28 | 0.46 | 1.12 | 1.24 | 3.04 | -0.08 |

Table 4: The results of fine-grained evaluation for Mix-56B on the MATH dataset based on topic and problem difficulty. n is the number of sample paths of the self-consistency, and t is the temperature. *AG*, *CP*, *GT*, *IA*, *NT*, *PA*, *PC* respectively represent *Algebra*, *Counting & Probability*, *Geometry*, *Intermediate Algebra*, *Number Theory*, *Prealgebra*, *Precalculus*. **Green** values indicate an performance improvement of HinSo prompting relative to CoT prompting, while **red** values indicate a decrease. Values in bold denote performance improvements greater than 1.

hints early can impact their planning process. Conversely, Standard and CoT prompting, focusing solely on the final answer or intermediate reasoning, are compatible with hints.

(2) *Larger model sizes tend to show more significant performance improvements.* From Tab. 8, we can observe that the average performance improvements for 7B, 13B, 56B, and 70B models across four prompting methods (e.g., CoT and LtM) are 0.5, 1.0, 1.3, and 1.5, respectively. The reason can be that the model capabilities increase as the size increases, and higher capabilities will help achieve higher quality hints for better problem-solving.

(3) *The introduction of HinSo can steadily enhance the performance of CoT prompting.* We observe that CoT, combined with HinSo, shows performance enhancements across all four models and six datasets, while SD, LtM, and PS all experience some scenarios of performance drop. From the line chart in Tab. 2, we can observe that LtM and PS exhibit significant fluctuations in average performance gains across each dataset, with numerous settings of negative improvement.

4.2 Q2: Can HinSo Work on Hard Tasks?

As the difficulty of the task increases, LLMs may not possess sufficient knowledge and capability to address it. This raises a research question: **Q2: Can LLMs generate helpful hints when they meet the challenge task?**

To answer this question, we chose to investigate the MATH dataset (Hendrycks et al., 2021b), a dataset that poses challenges for LLMs. The results are shown in Tab. 3. We can observe that only the Mix-56B model shows a significant improvement of 1.6 under CoT+HinSo prompting, while

the Llama-2 family model fails. The reason might be that the Llama-2 family models face significant challenges on the MATH dataset, with their best result being only 11.4 (Lm2-70B), while the Mix-56B model achieves 27.0 under CoT prompting, it is difficult for Llama-2 family model to generate valuable hints.

To find which kind of samples Mix-56B can work, we performed a fine-grained analysis based on the mathematic problem topic and the difficulty, where the dataset provides the topics and the difficulty levels. Furthermore, to explore how self-consistency affects the performance, we evaluate this model using sample paths of $n=4$ and $n=16$ and a model temperature of 0.4. The results are shown in Tab. 4. The main findings can be summarized as: (1) As n increases, under the CoT+HinSo setting, the samples for which the LLM sees performance improvements shift from low to high difficulty. (2) As n increases, it is commonly believed that the most challenging GT type experiences the most significant performance improvement, amounting to 4.17. These indicate that by increasing n , HinSo enhancement will correctly solve more complex questions.

4.3 Q3: How does SFT Perform on HinSo Format Datasets?

Despite the remarkable success of LLMs, most existing open-source LLMs (e.g., LLaMA-2) still face challenges in solving math problems due to complex reasoning processes. How do LLMs perform when they are supervised fine-tuning (SFT) on the HinSo format dataset?

| Model | Size | ACC | Model | Size | ACC |
|--------------------|------|-------------|------------------------------------|------|-------------|
| <i>open source</i> | | | <i>close source</i> | | |
| Llama2 | 7B | 14.6 | GPT-3.5 | - | 57.1 |
| Llama2 | 13B | 28.7 | PaLM | 540B | 56.5 |
| Llemma | 7B | 36.4 | Minerva | 540B | 58.8 |
| Llama2 | 34B | 42.2 | Minerva | 62B | 52.4 |
| RFT | 7B | 50.3 | Chinchilla | 70B | 43.7 |
| Llemma | 34B | 51.5 | <i>SFT on HST-S (7.5k samples)</i> | | |
| RFT | 13B | 54.8 | Llemma-CoT-S | 7B | 46.8 |
| WizardMath | 7B | 54.9 | Llemma-HinSo-S | 7B | 51.9 |
| WizardLM-V1.2 | 13B | 55.3 | <i>SFT on HST-L (75k samples)</i> | | |
| Llama2 | 70B | 56.8 | Llemma-CoT-L | 7B | 58.7 |
| WizardMath | 13B | 63.9 | Llemma-HinSo-L | 7B | 64.3 |

Table 5: The results of SFT on GSM8K. The values in **bold** denote best SFT result. The values in **blue** denote the mentioned baseline performance.

4.3.1 Training Dataset Collection

To make the LLMs learn the hint before solving the problem intuitively, we try to construct the high-quality and large-scale HinSo format dataset for conducting the supervised fine-tuning. We used GPT-4 to assist in constructing hints. Specifically, we selected the GSM8K training dataset covering 7,500 samples added a hint generated by guiding GPT-4 and manually verified to filter out or reconstruct samples with error in information. We build the HinSo-Training-Small (HST-S) with 7.5k training samples. Furthermore, we try to construct a larger training dataset. Based on MetaMathQA (Yu et al., 2023). We extracted 75k samples from MetaMathQA, which are derived from GSM8K and match the hint by found from HST-S. Finally, we created the HST-S with 7.5k samples and HinSo-Training-Large (HST-L) with 75k samples.

4.3.2 Baselines

The baselines considered include: Llama2 (Touvron et al., 2023c), RFT (Yuan et al., 2023), Llemma (Azerbayev et al., 2023), WizardMath (Luo et al., 2023b), WizardLM (Xu et al., 2023), GPT-3.5 (OpenAI, 2023), PaLM (Chowdhery et al., 2023), Minerva (Lewkowycz et al., 2022), and Chinchilla (Hoffmann et al., 2022)

4.3.3 Experimental Setup

We performed supervised fine-tuning on the Llemma-7B model using the constructed HinSo-formatted datasets, named HST-S (7.5k) and HST-L (75k). **Llemma-HinSo-S** and **Llemma-HinSo-L** are our models trained on the HST-S and HST-L, respectively. To ensure a fair comparison,

| Model | G8K | ASDiv | MArith | AQUA | MATH |
|-------------------------------------|--------------|--------------|--------------|--------------|--------------|
| Llemma# | 36.40 | - | - | - | 18.00 |
| Llemma-Infer | 40.11 | 69.58 | 89.43 | 33.47 | 17.02 |
| <i>SFT on CoT format datasets</i> | | | | | |
| Llemma-CoT-S | 46.80 | 69.62 | 90.27 | 35.43 | 17.88 |
| Llemma-CoT-L | 58.70 | 71.20 | 91.44 | 35.04 | 16.24 |
| <i>SFT on HinSo format datasets</i> | | | | | |
| Llemma-HinSo-S | 51.90 | 69.81 | 88.42 | 37.80 | 18.22 |
| Llemma-HinSo-L | 64.30 | 71.83 | 91.95 | 40.16 | 16.28 |

Table 6: Results of SFT models on mathematical reasoning datasets. # indicates results from the official Llemma report, while Llemma-Infer denotes results evaluated using the HinSo* prompt. Values in **bold** denote the best performance for each dataset.

we conduct the SFT using Llemma-7B on CoT-formatted HST-S and HST-L datasets, resulting in models **Llemma-CoT-S** and **Llemma-CoT-L**.

For the above SFT model based on CoT-formatted and HinSo-formatted datasets, the learning rate was set to $2 * 10^{-5}$, with a batch size of 64, over 5 epochs. For evaluation consistency, we set the temperature to 0. For the inference phrase, we evaluated **Llemma-CoT-S** and **Llemma-CoT-L** models using one demonstration sample for the GSM8K dataset.

4.3.4 Results and Analysis

The results with supervised fine-tuning on HST-S and HST-L datasets under Llemma-7B are shown in Tab. 5. The main observations are listed below: (1) *Supervised fine-tuning on datasets with HinSo format allows LLMs to achieve significant performance improvements.* From Tab. 5, we can observe that in three groups of SFT under CoT and HinSo dataset format, the performance dramatically improves with HinSo format, which is **5.1** (Llemma-CoT-S v.s Llemma-HinSo-S) and **5.6** (Llemma-CoT-L, Llemma-HinSo-L), respectively. The reason can be that SFT involving hints helps the model effectively utilize encoded knowledge during the reasoning stage, thereby improving the model’s generalization ability.

(2) *The result of Llemma-HinSo-L surpassed many popular LLMs, including GPT-3.5 and WizardMath.* By fine-tuning the HST-L dataset with 75k HinSo format samples, our Llemma-HinSo-L achieved a competitive performance of **64.3**, surpassing closed-source models such as GPT-3.5 (**57.1**) and PaLM-540B (**56.5**), and WizardMath-13B (**63.9**), which was fine-tuned on a large-scale

Question: Jill gets paid \$20 per hour to teach and \$30 to be a cheerleading coach. If she works 50 weeks a year, 35 hours a week as a teacher and 15 hours a week as a coach, what’s her annual salary? **True Answer:** 57500

CoT Response: Jill works 50 weeks a year. She works 35 hours a week as a teacher, so that’s $35 \times 50 = \$1750$ per year. She works 15 hours a week as a coach, so that’s $15 \times 50 = \$750$ per year. Her total annual salary is $\$1750 + \$750 = \$2500$. The answer is \$2500. [WRONG]

CoT+HinSo Response:

Hint: To find out Jill’s annual salary, **first calculate how much she makes per week as a teacher and as a coach. Then, multiply this total amount by the number of weeks she works in a year. This will give you Jill’s annual salary.**

Solution: Jill makes \$20 per hour as a teacher, so 35 hours a week is $35 \times 20 = 700$ per week. As a coach, she makes \$30 per hour, so 15 hours a week is $15 \times 30 = 450$ per week. $700 + 450 = 1150$ per week. 1150×50 weeks = 57500 per year. The answer is 57500. [CORRECT]

Table 7: Case studies of solving mathematical reasoning with CoT+HinSo and CoT prompting on the Mixtral-7*8B model. Blue text indicates the stem, pink text indicates the effective hint, cyan text indicates the judgment of whether the answer is correct, [CORRECT] denotes correct, and [WRONG] denotes incorrect.

459 mathematical corpus.

4.3.5 Generalization Capability Analysis

460 To further explore whether models fine-tuned
461 on HinSo-formatted datasets have better general-
462 ization capabilities, we conducted evaluation of
463 **Llemma-HinSo-S** and **Llemma-HinSo-L** on un-
464 seen datasets, namely ASDiv, MArith, AQUA, and
465 MATH. The results are shown in Tab. 6.

466 Findings: (1) *The generalization ability of mod-*
467 *els fine-tuned on HinSo-formatted datasets is supe-*
468 *rior to those fine-tuned on CoT-formatted datasets.*
469 This conclusion is supported by 7 out of 8 results
470 across two sets of models and four external datasets
471 (with the exception of the MArith dataset, where
472 the performance of Llemma-CoT-S is better than
473 Llemma-HinSo-S). (2) For the challenging dataset,
474 MATH, increasing the number of HinSo-formatted
475 training samples from GSM8K can’t improve per-
476 formance. This can be attributed to that SFT on
477 easier datasets may diminish Llemma-7B’s ability
478 to handle difficult datasets.
479

5 Further Analysis

480 How does guiding LLM to generate hints first affect
481 the generation of the model’s solution? We choose
482 to introduce hints under CoT prompting and select
483 case studies on mathematical reasoning, shown in
484 Tab 10. For more case analysis on common sense
485 reasoning can be seen in the appendix C.

486 In Tab 10, the solution from CoT seems reason-
487 able, but when calculating the annual total income
488 of a teacher and coach, it was not multiplied by
489 the hourly wage, leading to a final miscalculation.
490 In contrast, CoT+HinSo, within the hint, provided
491 the problem-solving ideas, allowing for the correct
492 answer to be calculated step by step in the solution
493 based on the problem-solving strategy mentioned
494 in the hint.
495

6 Related Work

496 Chain-of-thought (CoT) has given a lot of inspi-
497 ration to many works and has made numerous
498 attempts to explore high performance. These
499 techniques include using programming languages
500 to represent the reasoning process (Gao et al.,
501 2023; Lyu et al., 2023), representing the reason-
502 ing process with complex structures such as trees
503 or graphs (Yao et al., 2023a; Besta et al., 2023),
504 task decomposition (Zhou et al., 2022; Khot et al.,
505 2023) and combining different prompting (Liu
506 et al., 2023; Zhou et al., 2023b). For the use of
507 hint enhancement, Zheng et al. (2023) proposed
508 Progressive-Hint Prompting (PHP), which aims to
509 enhance LLMs’ effectiveness by introducing hints
510 iterative, where the hint is a numerical value ob-
511 tained from the previous solution (or base prompt’s
512 solution). However, the hints for our HinSo come
513 from LLMs themselves, while PHP comes from
514 previous predictions. Moreover, our hints can be
515 one-stage, whereas PHP must be multi-staged.
516

7 Conclusion

517 In this work, we present a Hint-before-Solving
518 framework (HinSo) to direct Large Language Mod-
519 els (LLMs) to initially produce hints that assist in
520 problem-solving before generating solutions that
521 incorporate intermediate reasoning steps. Through
522 extensive experiments, we have drawn several main
523 findings: (1) HinSo can guide LLMs to gener-
524 ate knowledge or key ideas for solving problems
525 (Sec. 4.1). (2) When meets challenging tasks,
526 HinSo fails on low-capability open-source LLMs
527 (e.g., Llama2-7B); however, while work on high-
528 capability open-source LLMs (Sec. 4.2). (3) SFT
529 on the HST-L dataset, our Llemma-HinSo-L (64.3)
530 improve a lot, outperforming GPT3.5 (57.1) and
531 WizardMath-13B (63.9) (Sec. 4.3).
532

533 Limitation

534 Here, we summarize some limitations of this paper,
535 as follows: (1) The HST-L dataset was expanded
536 by rewriting questions from GSM8K nine times,
537 but our hints were generated based only on the
538 original samples and applied to the nine rewritten
539 samples. The rewritten samples might undergo
540 logical changes, making the introduction of hints
541 less harmonious. There might be a risk of poor
542 performance during supervised fine-tuning. In the
543 future, we will refine this dataset carefully and
544 release a new version. (2) Due to limitations in
545 computational resources, this paper did not con-
546 duct supervised fine-tuning on models larger than
547 13B parameters in the SFT experiments, resulting
548 in an incomplete exploration of HinSo-enhanced
549 supervised fine-tuning. We will undertake this ex-
550 ploration in the future.

551 References

552 Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster,
553 Marco Dos Santos, Stephen McAleer, Albert Q.
554 Jiang, Jia Deng, Stella Biderman, and Sean Welleck.
555 2023. [Llemma: An open language model for mathe-](#)
556 [matics](#). *CoRR*, abs/2310.10631.

557 Maciej Besta, Nils Blach, Ales Kubicek, Robert Ger-
558 stenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz
559 Lehmann, Michal Podstawski, Hubert Niewiadom-
560 ski, Piotr Nyczyk, and Torsten Hoefer. 2023. [Graph](#)
561 [of thoughts: Solving elaborate problems with large](#)
562 [language models](#). *CoRR*, abs/2308.09687.

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

569 Aakanksha Chowdhery, Sharan Narang, Jacob Devlin,
570 Maarten Bosma, Gaurav Mishra, Adam Roberts,
571 Paul Barham, Hyung Won Chung, Charles Sutton,
572 Sebastian Gehrmann, Parker Schuh, Kensen Shi,
573 Sasha Tsvyashchenko, Joshua Maynez, Abhishek
574 Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vin-
575 odkumar Prabhakaran, Emily Reif, Nan Du, Ben
576 Hutchinson, Reiner Pope, James Bradbury, Jacob
577 Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin,
578 Toju Duke, Anselm Levskaya, Sanjay Ghemawat,
579 Sunipa Dev, Henryk Michalewski, Xavier Garcia,
580 Vedant Misra, Kevin Robinson, Liam Fedus, Denny
581 Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim,
582 Barret Zoph, Alexander Spiridonov, Ryan Sepassi,
583 David Dohan, Shivani Agrawal, Mark Omernick, An-
584 drew M. Dai, Thanumalayan Sankaranarayanan Pil-
585 lai, Marie Pellat, Aitor Lewkowycz, Erica Moreira,
586 Rewon Child, Oleksandr Polozov, Katherine Lee,

Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark
587 Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy
588 Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov,
589 and Noah Fiedel. 2023. [Palm: Scaling language mod-](#)
590 [eling with pathways](#). *J. Mach. Learn. Res.*, 24:240:1–
591 240:113. 592

Hyung Won Chung, Le Hou, Shayne Longpre, Barret
593 Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang,
594 Mostafa Dehghani, Siddhartha Brahma, Albert Web-
595 son, Shixiang Shane Gu, Zhuyun Dai, Mirac Suz-
596 gun, Xinyun Chen, Aakanksha Chowdhery, Sharan
597 Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao,
598 Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav
599 Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam
600 Roberts, Denny Zhou, Quoc V. Le, and Jason Wei.
601 2022. [Scaling instruction-finetuned language models](#).
602 *CoRR*, abs/2210.11416. 603

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
604 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
605 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
606 Nakano, et al. 2021. Training verifiers to solve math
607 word problems. *arXiv e-prints*, pages arXiv–2110. 608

Roi Cohen, Mor Geva, Jonathan Berant, and Amir
609 Globerson. 2023. [Crawling the internal knowledge-](#)
610 [base of language models](#). In *Findings of the Associ-*
611 *ation for Computational Linguistics: EACL 2023,*
612 *Dubrovnik, Croatia, May 2-6, 2023*, pages 1811–
613 1824. Association for Computational Linguistics. 614

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and
615 Tushar Khot. 2023. [Complexity-based prompting for](#)
616 [multi-step reasoning](#). In *The Eleventh International*
617 *Conference on Learning Representations, ICLR 2023,*
618 *Kigali, Rwanda, May 1-5, 2023*. OpenReview.net. 619

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon,
620 Pengfei Liu, Yiming Yang, Jamie Callan, and Gra-
621 ham Neubig. 2023. [PAL: program-aided language](#)
622 [models](#). In *International Conference on Machine*
623 *Learning, ICML 2023, 23-29 July 2023, Honolulu,*
624 *Hawaii, USA*, volume 202 of *Proceedings of Machine*
625 *Learning Research*, pages 10764–10799. PMLR. 626

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot,
627 Dan Roth, and Jonathan Berant. 2021. [Did aristotle](#)
628 [use a laptop? A question answering benchmark with](#)
629 [implicit reasoning strategies](#). *Trans. Assoc. Comput.*
630 *Linguistics*, 9:346–361. 631

Hangfeng He, Hongming Zhang, and Dan Roth. 2023.
632 [Rethinking with retrieval: Faithful large language](#)
633 [model inference](#). *CoRR*, abs/2301.00303. 634

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul
635 Arora, Steven Basart, Eric Tang, Dawn Song, and
636 Jacob Steinhardt. 2021a. Measuring mathematical
637 problem solving with the math dataset. In *Thirty-*
638 *fifth Conference on Neural Information Processing*
639 *Systems Datasets and Benchmarks Track (Round 2)*. 640

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul
641 Arora, Steven Basart, Eric Tang, Dawn Song, and
642 Jacob Steinhardt. 2021b. [Measuring mathematical](#)
643

| | | | |
|-----|--|--|---|
| 644 | problem solving with the MATH dataset . In <i>Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual</i> . | Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023. Plan, verify and switch: Integrated reasoning with diverse x-of-thoughts . In <i>Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023</i> , pages 2807–2822. Association for Computational Linguistics. | 703 704 705 706 707 708 709 710 |
| 649 | Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models . <i>CoRR</i> , abs/2203.15556. | Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023. A survey of deep learning for mathematical reasoning . In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023</i> , pages 14605–14631. Association for Computational Linguistics. | 711 712 713 714 715 716 717 |
| 658 | Yushi Hu, Chia-Hsuan Lee, Tianbao Xie, Tao Yu, Noah A. Smith, and Mari Ostendorf. 2022. In-context learning for few-shot dialogue state tracking . In <i>Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022</i> , pages 2627–2643. Association for Computational Linguistics. | Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023a. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct . <i>CoRR</i> , abs/2308.09583. | 718 719 720 721 722 723 |
| 665 | Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models . In <i>Proceedings of the The 61st Annual Meeting of the Association for Computational Linguistics: Industry Track, ACL 2023, Toronto, Canada, July 9-14, 2023</i> , pages 37–42. Association for Computational Linguistics. | Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023b. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct . <i>CoRR</i> , abs/2308.09583. | 724 725 726 727 728 729 |
| 672 | Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks . In <i>The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023</i> . OpenReview.net. | Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning . <i>CoRR</i> , abs/2301.13379. | 730 731 732 733 |
| 679 | Zachary Levonian, Chenglu Li, Wangda Zhu, Anoushka Gade, Owen Henkel, Millie-Ellen Postle, and Wanli Xing. 2023. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference . <i>CoRR</i> , abs/2310.03184. | Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers . <i>CoRR</i> , abs/2106.15772. | 734 735 736 737 |
| 685 | Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models . In <i>Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022</i> . | Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In <i>Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022</i> , pages 11048–11064. Association for Computational Linguistics. | 738 739 740 741 742 743 744 745 746 |
| 695 | Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems . In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers</i> , pages 158–167. Association for Computational Linguistics. | Mistral AI Team. 2023. Mixtral of experts . https://mistral.ai/news/mixtral-of-experts/ . Accessed: 2023-12-26. | 747 748 749 750 |
| 702 | | OpenAI. 2023. GPT-4 technical report . <i>CoRR</i> , abs/2303.08774. | 751 752 |
| | | Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks . In <i>Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021</i> , volume ACL/IJCNLP 2021 of <i>Findings of ACL</i> , | 753 754 755 756 757 758 759 |

| | | | |
|-----|--|--|------|
| 760 | pages 4179–4192. Association for Computational Linguistics. | | 818 |
| 761 | | | 819 |
| 762 | Subhro Roy and Dan Roth. 2016. Solving general arithmetic word problems . <i>CoRR</i> , abs/1608.01413. | | 820 |
| 763 | | | 821 |
| 764 | Maarten Sap, Vered Shwartz, Antoine Bosselut, Yejin Choi, and Dan Roth. 2020. Commonsense reasoning for natural language processing . In <i>Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2020, Online, July 5, 2020</i> , pages 27–33. Association for Computational Linguistics. | | 822 |
| 765 | | | 823 |
| 766 | | | 824 |
| 767 | | | 825 |
| 768 | | | 826 |
| 769 | | | 827 |
| 770 | | | 828 |
| 771 | Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. <i>arXiv preprint arXiv:2206.04615</i> . | | 829 |
| 772 | | | 830 |
| 773 | | | 831 |
| 774 | | | 832 |
| 775 | | | 833 |
| 776 | | | 834 |
| 777 | | | 835 |
| 778 | Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models . <i>CoRR</i> , abs/2302.13971. | | 836 |
| 779 | | | 837 |
| 780 | | | 838 |
| 781 | | | 839 |
| 782 | | | 840 |
| 783 | | | 841 |
| 784 | | | 842 |
| 785 | Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models . <i>CoRR</i> , abs/2307.09288. | | 843 |
| 786 | | | 844 |
| 787 | | | 845 |
| 788 | | | 846 |
| 789 | | | 847 |
| 790 | | | 848 |
| 791 | | | 849 |
| 792 | | | 850 |
| 793 | | | 851 |
| 794 | | | 852 |
| 795 | | | 853 |
| 796 | | | 854 |
| 797 | | | 855 |
| 798 | | | 856 |
| 799 | | | 857 |
| 800 | | | 858 |
| 801 | | | 859 |
| 802 | | | 860 |
| 803 | | | 861 |
| 804 | | | 862 |
| 805 | | | 863 |
| 806 | | | 864 |
| 807 | | | 865 |
| 808 | Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, et al. 2023c. Llama 2: Open foundation and fine-tuned chat models . <i>arXiv preprint arXiv:2307.09288</i> . | | 866 |
| 809 | | | 867 |
| 810 | | | 868 |
| 811 | | | 869 |
| 812 | | | 870 |
| 813 | | | 871 |
| 814 | Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023a. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language | | 872 |
| 815 | | | 873 |
| 816 | | | 874 |
| 817 | | | 875 |
| | | | 876 |
| | | | 877 |
| | | | 878 |
| | | | 879 |
| | | | 880 |
| | | | 881 |
| | | | 882 |
| | | | 883 |
| | | | 884 |
| | | | 885 |
| | | | 886 |
| | | | 887 |
| | | | 888 |
| | | | 889 |
| | | | 890 |
| | | | 891 |
| | | | 892 |
| | | | 893 |
| | | | 894 |
| | | | 895 |
| | | | 896 |
| | | | 897 |
| | | | 898 |
| | | | 899 |
| | | | 900 |
| | | | 901 |
| | | | 902 |
| | | | 903 |
| | | | 904 |
| | | | 905 |
| | | | 906 |
| | | | 907 |
| | | | 908 |
| | | | 909 |
| | | | 910 |
| | | | 911 |
| | | | 912 |
| | | | 913 |
| | | | 914 |
| | | | 915 |
| | | | 916 |
| | | | 917 |
| | | | 918 |
| | | | 919 |
| | | | 920 |
| | | | 921 |
| | | | 922 |
| | | | 923 |
| | | | 924 |
| | | | 925 |
| | | | 926 |
| | | | 927 |
| | | | 928 |
| | | | 929 |
| | | | 930 |
| | | | 931 |
| | | | 932 |
| | | | 933 |
| | | | 934 |
| | | | 935 |
| | | | 936 |
| | | | 937 |
| | | | 938 |
| | | | 939 |
| | | | 940 |
| | | | 941 |
| | | | 942 |
| | | | 943 |
| | | | 944 |
| | | | 945 |
| | | | 946 |
| | | | 947 |
| | | | 948 |
| | | | 949 |
| | | | 950 |
| | | | 951 |
| | | | 952 |
| | | | 953 |
| | | | 954 |
| | | | 955 |
| | | | 956 |
| | | | 957 |
| | | | 958 |
| | | | 959 |
| | | | 960 |
| | | | 961 |
| | | | 962 |
| | | | 963 |
| | | | 964 |
| | | | 965 |
| | | | 966 |
| | | | 967 |
| | | | 968 |
| | | | 969 |
| | | | 970 |
| | | | 971 |
| | | | 972 |
| | | | 973 |
| | | | 974 |
| | | | 975 |
| | | | 976 |
| | | | 977 |
| | | | 978 |
| | | | 979 |
| | | | 980 |
| | | | 981 |
| | | | 982 |
| | | | 983 |
| | | | 984 |
| | | | 985 |
| | | | 986 |
| | | | 987 |
| | | | 988 |
| | | | 989 |
| | | | 990 |
| | | | 991 |
| | | | 992 |
| | | | 993 |
| | | | 994 |
| | | | 995 |
| | | | 996 |
| | | | 997 |
| | | | 998 |
| | | | 999 |
| | | | 1000 |

| | | | |
|-----|--|---|-----|
| 874 | Mammoth: Building math generalist models through hybrid instruction tuning. <i>CoRR</i> , abs/2309.05653. | A When HinSo Meets Existing Prompting Methods | 914 |
| 875 | | | 915 |
| 876 | Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023a. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. | Fig. 4 shows the examples of input and output before (four examples at the top) and after (four examples at the bottom) applying HinSo to standard Least-to-Most, Plan-and-Solve, and CoT promptings. | 916 |
| 877 | In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , <i>ACL 2023, Toronto, Canada, July 9-14, 2023</i> , pages 5823–5840. Association for Computational Linguistics. | | 917 |
| 878 | | | 918 |
| 879 | | | 919 |
| 880 | | | 920 |
| 881 | | | |
| 882 | | B Experiments and Results | 921 |
| 883 | | | |
| 884 | Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023b. A survey of large language models. <i>CoRR</i> , abs/2303.18223. | B.1 Q1: Can HinSo Work? | 922 |
| 885 | | | |
| 886 | | B.1.1 Effectiveness of HinSo for CoT Prompting | 923 |
| 887 | | | 924 |
| 888 | | In Exp-I, we found that applying HinSo to CoT prompting results in significant and stable performance improvements across six datasets. Based on this, to identify flexible and effective ways to incorporate HinSo, we attempted to explore whether a two-stage HinSo (HinSo2) approach could work in CoT prompting. The two-stage HinSo means that LLMs produce outputs twice, first outputting a hint and then a solution. In contrast, HinSo has only one output that contains both the hint and the solution. Experimental results on 6 datasets of 4 open source models are shown in Tab. 8. The main observations are summarized as below: | 925 |
| 889 | | (1) <i>The performance of HinSo and HinSo2 is comparable, despite the different ways of introducing hints.</i> We can observe that among four LLMs, the largest average performance gap between HinSo and HinSo2 across six datasets was achieved on the Llama2-13B model with 0.5% (56.7-56.2). This indicates that although the methods of introducing hints differ, the extent of performance improvement brought by both is close. | 926 |
| 890 | | (2) <i>HinSo brings more stable improvements compared to HinSo2.</i> From histograms in Tab. 8, HinSo shows improvements on nearly every dataset under models of four different sizes. In contrast, HinSo2 may lead to performance decreases in certain scenarios, for example, on the MArith dataset, the HinSo2 performance decreases with Llama2-7B and Llama2-70B models. | 927 |
| 891 | | | 928 |
| 892 | Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. <i>CoRR</i> , abs/2304.09797. | | 929 |
| 893 | | | 930 |
| 894 | | | 931 |
| 895 | | | 932 |
| 896 | Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. <i>arXiv preprint arXiv:2205.10625</i> . | | 933 |
| 897 | | | 934 |
| 898 | | | 935 |
| 899 | | | 936 |
| 900 | | | 937 |
| 901 | | | 938 |
| 902 | Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. 2023a. Least-to-most prompting enables complex reasoning in large language models. In <i>The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023</i> . OpenReview.net. | | 939 |
| 903 | | | 940 |
| 904 | | | 941 |
| 905 | | | 942 |
| 906 | | | 943 |
| 907 | | | 944 |
| 908 | | | 945 |
| 909 | | | 946 |
| 910 | Jianpeng Zhou, Wanjun Zhong, Yanlin Wang, and Jiahai Wang. 2023b. Adaptive-solver framework for dynamic strategy selection in large language model reasoning. <i>CoRR</i> , abs/2310.01446. | | 947 |
| 911 | | | 948 |
| 912 | | | 949 |
| 913 | | | 950 |
| | | | 951 |
| | | | 952 |
| | | | 953 |
| | | | 954 |
| | | B.1.2 The Impact of Hint Quality | 955 |
| | | Introducing HinSo can effectively enhance the performance of CoT prompting. But what is the upper bound? Here, we choose to explore on HinSo2 because it enables the hints from external sources, a feature not available in the one-stage HinSo structure, and HinSo2 is comparable in strength to HinSo (Sec. B.1.1). Hints generated by GPT-4 | 956 |
| | | | 957 |
| | | | 958 |
| | | | 959 |
| | | | 960 |
| | | | 961 |
| | | | 962 |

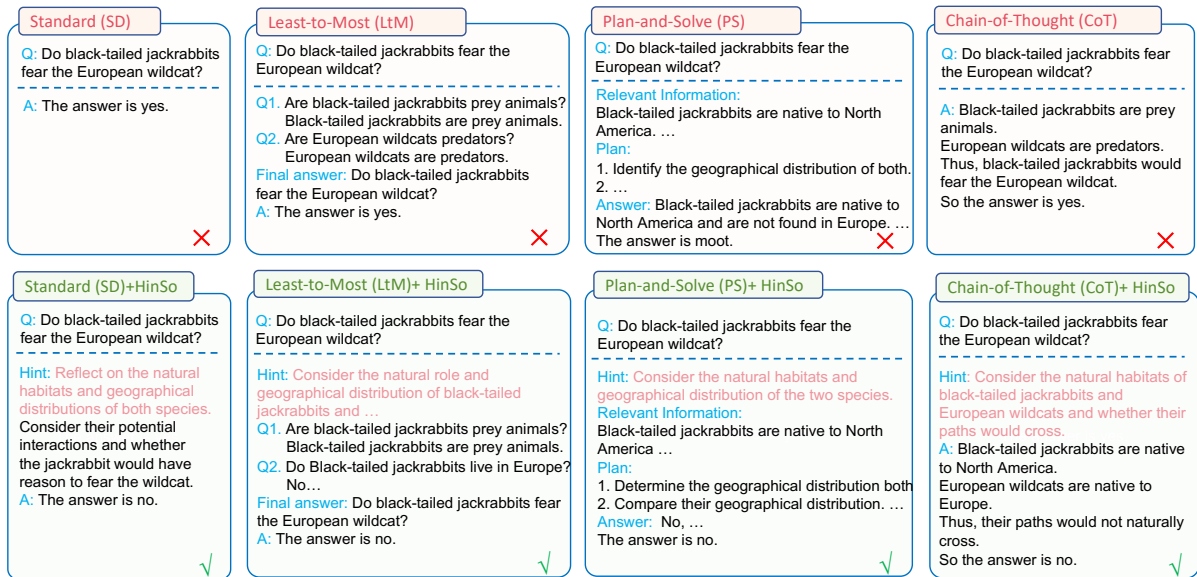


Figure 4: Examples of input and output before (four examples at the top) and after (four examples at the bottom) applying HinSo to standard Least-to-Most, Plan-and-Solve, and CoT promptings. The red text in the textbox indicates hints. We find that hints from LLMs, including problem-solving ideas close to the correct answer (e.g., geographical distributions of both species), guide LLMs to use accurate knowledge for correct and logical reasoning.

| Method | | G8K | ASDiv | MArith | AQUA | SQA | Date | Avg | Improvement |
|---------|----------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Lm2-7B | CoT | 19.7 | 53.6 | 63.4 | 24.4 | 66.3 | 40.1 | 44.6 | |
| | +HinSo [†] | 19.9 | 55.8 | 63.8 | 24.4 | 67.5 | 43.2 | 45.8 | |
| | +HinSo2 [‡] | 22.6 | 55.4 | 62.6 | 25.2 | 66.8 | 40.4 | 45.5 | |
| Lm2-13B | CoT | 34.5 | 60.5 | 83.2 | 25.6 | 68.0 | 57.7 | 54.9 | |
| | +HinSo [†] | 36.5 | 61.2 | 87.1 | 25.6 | 72.1 | 57.7 | 56.7 | |
| | +HinSo2 [‡] | 36.5 | 61.9 | 85.1 | 26.0 | 70.7 | 57.0 | 56.2 | |
| Lm2-70B | CoT | 46.1 | 72.5 | 93.8 | 35.8 | 74.6 | 71.6 | 65.7 | |
| | +HinSo [†] | 50.3 | 74.4 | 94.6 | 37.0 | 77.0 | 73.0 | 67.7 | |
| | +HinSo2 [‡] | 54.3 | 73.9 | 93.0 | 37.8 | 71.5 | 76.0 | 67.8 | |
| Mix-56B | CoT | 63.7 | 78.3 | 96.1 | 42.5 | 74.7 | 69.9 | 70.9 | |
| | +HinSo [†] | 69.3 | 80.1 | 97.0 | 48.4 | 75.1 | 77.4 | 74.6 | |
| | +HinSo2 [‡] | 69.8 | 80.3 | 96.8 | 45.7 | 74.3 | 79.1 | 74.3 | |

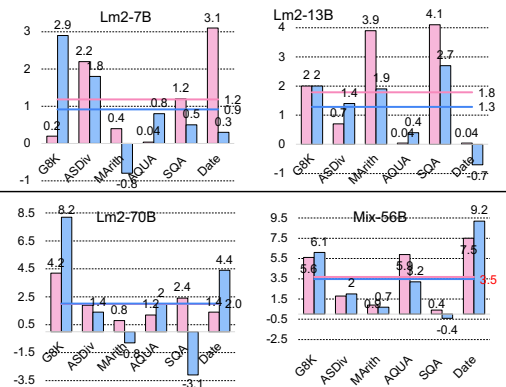


Table 8: The results of applying HinSo and HinSo2 in CoT prompting. The **bold** values indicate the best performance. [†] and [‡] denote that the performance of HinSo and HinSo2 is significantly better than CoT prompting, respectively.

will be used as part of the input in the HinSo2, denoting as HinSo2G. Experimental results are shown in Tab. 9. The performance of ChatGPT is copied from Yin et al. (2023), where the number of examples used to evaluate GSM8K, MultiArith, and AQUA is 8, 8, and 4, respectively. The main findings are summarized as below:

(1) *High-quality hints make the open-source model outperforms ChatGPT.* We can observe that with the introduction of high-quality hints, all of the four LLMs with different model sizes and structures consistently showed performance improvement across six datasets. Furthermore, the Mix-56B equipped with HinSo2(GPT4) outperformed ChatGPT on the GSM8K, MultiArith, and AQUA

datasets.

(2) *The introduction of high-quality hints leads to more improvements in lower-capability models.* Tab. 9 shows that the average performance improvements for the Llama2 models sized 7B, 13B, and 70B are 12.8, 9.9, and 7.7, respectively. This indicates that with the support of high-quality hints, HinSo2(GPT4)’s performance has improved a lot compared to HinSo2. This can be attributed to that the low capability LLMs are hard to generate helpful hints that can assist in providing correct solutions. By providing high-quality hints, it is possible to offer more benefits beyond the capability of lower-ability LLMs. Therefore, there is a relatively large improvement in performance.

| Method | G8K | ASDiv | MArith | AQUA | SQA | Date | Avg | |
|----------|---------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| ChatGPT | 79.1 | - | 97.3 | 55.1 | - | - | - | |
| 7B | HinSo2 | 22.6 | 55.4 | 62.6 | 25.2 | 66.8 | 40.4 | 45.5 |
| | HinSo2G | 39.0 | 62.5 | 88.9 | 28.7 | 69.5 | 61.0 | 58.3 |
| | Impv | 16.4 | 7.1 | 26.3 | 3.5 | 2.7 | 20.6 | 12.8 |
| 13B | HinSo2 | 36.5 | 61.9 | 85.1 | 26.0 | 70.7 | 57.0 | 56.2 |
| | HinSo2G | 56.4 | 66.4 | 95.6 | 36.6 | 72.0 | 69.4 | 66.1 |
| | Impv | 19.9 | 4.5 | 10.5 | 10.6 | 1.3 | 12.4 | 9.9 |
| 70B | HinSo2 | 54.3 | 73.9 | 93.0 | 37.8 | 71.5 | 76.0 | 67.8 |
| | HinSo2G | 68.2 | 79.0 | 98.0 | 43.3 | 76.6 | 87.7 | 75.5 |
| | Impv | 13.9 | 5.1 | 5.0 | 5.5 | 5.1 | 11.7 | 7.7 |
| 56B | HinSo2 | 69.8 | 80.3 | 96.8 | 45.7 | 74.3 | 79.1 | 74.3 |
| | HinSo2G | 79.5 | 84.1 | 99.2 | 56.3 | 76.5 | 84.7 | 80.1 |
| | Impv | 9.7 | 3.8 | 2.4 | 10.6 | 2.2 | 5.6 | 5.7 |
| Avg impv | 15.0 | 5.1 | 11.1 | 7.6 | 2.8 | 12.6 | 9.0 | |

Table 9: Experimental results of enhancing HinSo2 with hints generated by GPT4. The values in green are the performance gap between HinSo2G and HinSo2. The blue values are the improvement across the four models. The values in bold represent the best performance.

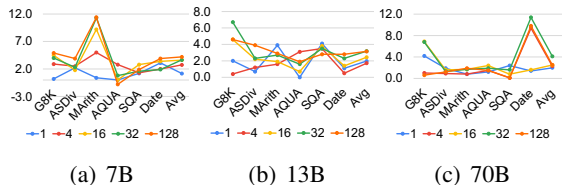


Figure 5: The relative performance improvement of self-consistency between CoT+HinSo and CoT. The numbers of sample paths are 4, 16, 32, and 128, and the model temperature is 0.4.

B.2 Q2: Can HinSo Work on Hard Tasks?

B.2.1 The Impact of Self-consistency

In EXP-IV (Sec. 4.2), we found that self-consistency setting can improve performance of difficult tasks (MATH dataset), even difficult samples. This raises the question of how CoT prompting equipped with HinSo performs under a self-consistency setting for the popular tasks. We sample paths with numbers (n) 4, 16, 32, and 128 for the self-consistency study and set the model temperature as 0.4. The relative improvement between CoT+HinSo and CoT on six datasets is shown in Fig. 5 (Full results can be seen in the Appendix G). The main findings are as below:

(1) *As the number of sampling paths increases, the relative improvements brought by applying HinSo also increase.* From Fig. 5, we can observe that at n=32 or n=128, all three models achieve their best performance. By calculating the Pearson correlation between the number of sampling (n) and relative performance for Lm2-7B, Lm2-13B, and Lm2-70B (excluding n=128), the correlations are 0.67, 0.72, and 0.95, respectively. The reason can

be that the larger n leads to more explored hints, making it easier to generate hints beneficial for problem-solving.

(2) *Smaller models see the most significant relative performance improvement after applying self-consistency.* This might be because smaller models have lower capabilities, while with guided hints, increasing n makes it easier to correct originally incorrect solutions, thus leading to more substantial performance improvements.

C Analysis

C.1 Length of Reasoning

Can HinSo enhance the model’s reasoning capability and effectively reduce the length of the solution generated? To answer this question, we calculated the solution lengths for CoT and CoT+HinSo (applying HinSo to CoT). For easy understanding, we divided the solution length of CoT+HinSo by the solution length of CoT, with the results shown in Fig. 6, where the red horizontal line indicates that the solution lengths of CoT and CoT+HinSo are equal.

Our main observation are summarized as below:

(1) *Introducing HinSo can effectively reduce the length of the solution.* From Fig. 6, we can observe that, out of 24 results across four models and six datasets, only 5 instances show CoT+HinSo having a longer solution length than CoT.

(2) *The effect of reducing the solution length by introducing HinSo is most pronounced in mathematical reasoning tasks.*

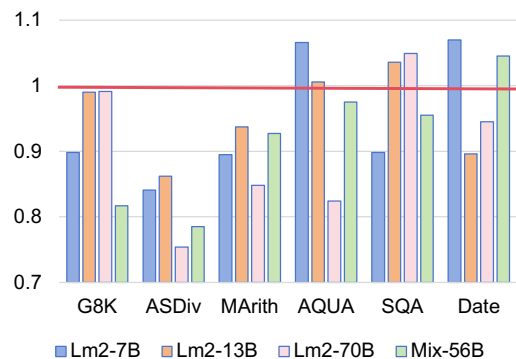


Figure 6: The ratio of solution lengths between CoT and HinSo+CoT (HinSo applied to CoT prompting). The red line ($y=1$) indicates that the solution lengths of CoT equals to HinSo+CoT.

1047
1048
1049
1050
1051
1052
1053
1054
1055

1056
1057
1058
1059
1060
1061
1062
1063
1064
1065

1066
1067
1068
1069
1070
1071
1072
1073
1074

1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095

C.2 Case Study

Guiding the model to generate hints before the solution can effectively improve the model’s performance. So, how does guiding LLM to generate hints first affect the generation of the model’s solution? We choose to introduce hints under CoT prompting and select case studies on mathematical reasoning and common sense reasoning tasks, as shown in Tab 10.

Case 1 For the question, "Could a Jujutsu expert hypothetically defeat a Janissary?". Under CoT prompting, the LLM-generated solution only explained what "Jujutsu expert" and "Janissary" are. However, in CoT+HinSo, the generated hint mentioned analyzing the possibility of the Jujutsu expert defeating Janissary from the perspectives of "martial arts skills" and "weapons," thus making a correct solution followed up after introducing the hint.

Case 2 The solution from CoT seems reasonable, but when calculating the annual total income of a teacher and coach, it was not multiplied by the hourly wage, leading to a final miscalculation. In contrast, CoT+HinSo, within the hint, provided the problem-solving ideas, allowing for the correct answer to be calculated step by step in the solution based on the problem-solving strategy mentioned in the hint.

C.3 Robustness Analysis

Considering the impact that varying sets of examples may have on results, the question arises: Is the HinSo framework effective with diverse example sets?

To investigate this, we conducted experiments on the GSM8K (mathematical reasoning) and StrategyQA (common sense reasoning) datasets. Like the setting in Exp-I, we randomly chose four sets of examples from the testing set, each comprising 8 examples for GSM8K and 6 examples for StrategyQA. We then crafted hints and solutions featuring intermediate reasoning steps aided by GPT-4. These experiments were carried out on four LLMs: Llama2-7B, Llama2-13B, Llama2-70B, and Mixtral-8*7B. According to the results presented in Tab. 11, CoT+HinSo consistently outperformed CoT across the GSM8K and StrategyQA datasets, with all four models showing significant performance enhancements across the four example sets. This demonstrates the robustness of the

performance gains achieved by integrating CoT with HinSo.

D Prompt Example

The four models evaluated in this paper, namely Lm2-7B, Lm2-13B, Lm2-70B, and Mix-56B, were all tested using the same prompt template. Tab. 12 shows the prompt template for mathematical reasoning and common sense reasoning tasks.

Tab. 13 shows the prompt template for the GPT4 to generate hints for constructing our HST-S and HST-L training datasets.

E Case Study

Guiding the model to generate hints before the solution can effectively improve the model’s performance. So, how does guiding LLM to generate hints first affect the generation of the model’s solution? We choose to introduce hints under CoT prompting and select case studies on mathematical reasoning and common sense reasoning tasks, as shown in Tab 14.

Case 1 The solution from CoT appears logical, but its analysis remains superficial, merely focusing on the relationship between the two entities (US brand Nice and the Western honey bee) to answer the question. In contrast, the hint from CoT+HinSo suggests approaching from a deeper view, specifically questioning whether the crops relied upon by US brand Nice depend on Western honey bees for pollination, which leads to the correct answer.

Case 2 For question "Do black-tailed jackrabbits fear the European wildcat?", CoT only considered the biological perspective, leading to an incorrect answer. However, the hint from CoT+HinSo suggested that it is necessary to consider not only the biological aspect but also the habitat of the organism, thereby achieving the correct answer.

Case 3 We can observe that CoT’s calculation method overlooks an important piece of knowledge, namely the formula for calculating the perimeter: "The distance traveled by a point on the edge of a rotating object equals the circle’s circumference.". In contrast, CoT+Hint successfully suggests utilizing the formula for perimeter, thereby obtaining the correct answer.

Case 4 The question involves calculating the perimeter of a rectangle, but the CoT method only adds the width and height of a rectangle.

1096
1097

1098

1099
1100
1101
1102
1103
1104
1105
1106

1107

1108
1109
1110
1111
1112
1113
1114
1115

1116
1117
1118
1119
1120
1121
1122
1123
1124

1125
1126
1127
1128
1129
1130
1131

1132
1133
1134
1135
1136
1137
1138
1139

1140
1141
1142

Case 1: Commonsense reasoning

Question: Could a Jujutsu expert hypothetically defeat a Janissary?

True Answer: FALSE

CoT Response: Jujutsu is a martial art that focuses on using an opponent’s strength against them. Janissaries were elite soldiers in the Ottoman Empire. Thus, a Jujutsu expert could potentially defeat a Janissary. So the answer is yes. [WRONG]

CoT+HinSo Response:

Hint: Compare the martial arts skills of a Jujutsu expert with the combat training of a Janissary, considering the historical context and weaponry.

Solution: A Jujutsu expert has martial arts skills, but no weapons. A Janissary is an elite Ottoman soldier with advanced combat training and weaponry. Thus, a Janissary would likely defeat a Jujutsu expert. So the answer is no. [CORRECT]

Case 2: Mathematical reasoning

Question: Jill gets paid \$20 per hour to teach and \$30 to be a cheerleading coach. If she works 50 weeks a year, 35 hours a week as a teacher and 15 hours a week as a coach, what’s her annual salary?

True Answer: 57500

CoT Response: Jill works 50 weeks a year. She works 35 hours a week as a teacher, so that’s $35 \times 50 = \$1750$ per year. She works 15 hours a week as a coach, so that’s $15 \times 50 = \$750$ per year. Her total annual salary is $\$1750 + \$750 = \$2500$. The answer is \$2500. [WRONG]

CoT+HinSo Response:

Hint: To find out Jill’s annual salary, first calculate how much she makes per week as a teacher and as a coach. Then, multiply this total amount by the number of weeks she works in a year. This will give you Jill’s annual salary.

Solution: Jill makes \$20 per hour as a teacher, so 35 hours a week is $35 \times 20 = 700$ per week. As a coach, she makes \$30 per hour, so 15 hours a week is $15 \times 30 = 450$ per week. $700 + 450 = 1150$ per week. 1150×50 weeks = 57500 per year. The answer is 57500. [CORRECT]

Table 10: Case studies of solving mathematical reasoning and common sense reasoning problems with CoT+HinSo and CoT prompting on the Mixtral-7*8B model. Blue text indicates the stem, pink text indicates the effective hint, cyan text indicates the judgment of whether the answer is correct, [CORRECT] denotes correct, and [WRONG] denotes incorrect.

| Model | HinSo | GSM8K | | | | SQA | | | |
|---------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | E1 | E2 | E3 | E4 | E1 | E2 | E3 | E4 |
| Lm2-7B | × | 20.2 | 15.2 | 18.0 | 17.0 | 61.2 | 56.6 | 63.9 | 60.9 |
| | ✓ | 22.7 | 21.6 | 23.4 | 22.8 | 63.8 | 61.5 | 65.9 | 63.3 |
| Lm2-13B | × | 35.9 | 29.1 | 25.4 | 32.2 | 64.1 | 60.6 | 67.5 | 63.2 |
| | ✓ | 37.1 | 34.7 | 35.1 | 36.5 | 67.4 | 62.0 | 68.2 | 65.9 |
| Lm2-70B | × | 53.7 | 54.1 | 54.4 | 54.0 | 71.1 | 65.1 | 75.1 | 68.2 |
| | ✓ | 60.1 | 56.3 | 55.3 | 59.3 | 71.7 | 72.1 | 75.8 | 73.1 |
| Lm2-56B | × | 67.9 | 68.8 | 67.2 | 67.8 | 65.4 | 60.3 | 69.3 | 61.9 |
| | ✓ | 69.1 | 69.1 | 68.2 | 68.8 | 67.3 | 64.5 | 70.6 | 66.8 |

Table 11: Experimental results for CoT Prompting with and without HinSo on the GSM8K and StrategyQA (SQA) datasets across various example groups (E1, E2, E3, and E4). Values in bold denote the best results.

CoT+HinSo suggested that the perimeter be calculated by four lengths, making the final answer calculation correct.

1143
1144
1145

F Reference Baseline

1146

In this paper, we reimplemented the results of four models, namely Llama-7B, Llama-13B, Llama-70B, and Mixtral-7*8B, under SD, LtM, PS, and CoT promptings, to compare with our HinSo-enhanced promptings’ performance. Are our reimplemented results within a reasonable range? To answer this question, we compared our reimplemented results with results from some recently works across six datasets: GSM8K, AQUA, ASDiv, Date, MultiArith, and StrategyQA. The results are shown in Fig. 7.

1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157

There is a considerable amount of existing work on CoT prompting, while results for SD, LtM, and PS prompting are limited. The baseline work we present in the Fig. 7 comes from five studies that cover a broad range of baseline methods. We can observe that across these six datasets, except for Llama-7B, which often lacks a closely matched model size for a baseline, the results for Llama-13B, Llama-70B, and Mixtral-7*8B are compara-

1158
1159
1160
1161
1162
1163
1164
1165
1166

| Mathematical reasoning |
|---|
| <p>Please answer the following question.</p> <p>Example 1: Question: Shawn has five toys. For Christmas, he got two toys each from his mom and dad. How many toys does he have now?</p> <p>Hint: Begin with the number of toys Shawn had initially. Then, add the number of toys he received from each parent. Remember, each parent gave him a certain number of toys, so you'll need to add those to his original amount to find out how many toys he has now.</p> <p>Solution: Shawn started with 5 toys. If he got 2 toys each from his mom and dad, then that is 4 more toys. $5 + 4 = 9$. The answer is 9.</p> <p>..... (Omitting 7 examples)</p> <p>Testing Example: Question: [QUESTION]</p> |
| Commonsense reasoning |
| <p>Please answer the following question.</p> <p>Example 1: Question: Do hamsters provide food for any animals?</p> <p>Hint: Consider the natural role of hamsters in the food chain and who might rely on them as a source of nutrition.</p> <p>Solution: Hamsters are prey animals. Prey are food for predators. Thus, hamsters provide food for some animals. So the answer is yes.</p> <p>..... (Omitting 5 examples)</p> <p>Testing Example: Question: [QUESTION]</p> |

Table 12: Prompt template for the evaluation of LLMs based on the HinSo framework on mathematical reasoning and commonsense reasoning.

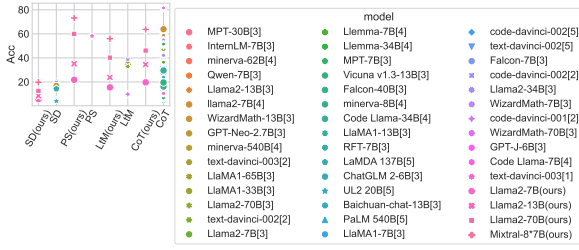
| Mathematical reasoning |
|--|
| <p>Please generate a hint for solving the following question.</p> <p>Example 1: Question: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?</p> <p>Hint: Think about how many trees there were at the beginning and how many there are at the end. To find out how many trees were planted, you need to figure out the difference between these two numbers.</p> <p>..... (Omitting 7 examples)</p> <p>Testing Example: Question: [QUESTION]</p> |

Table 13: A prompt template for the GPT4 generates the hint for constructing our HST-S and HST-L training datasets.

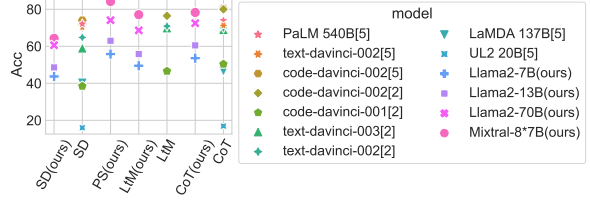
1167 ble to some existing open-source or closed-source
1168 models.

1169 **G Results of Self-consistency**

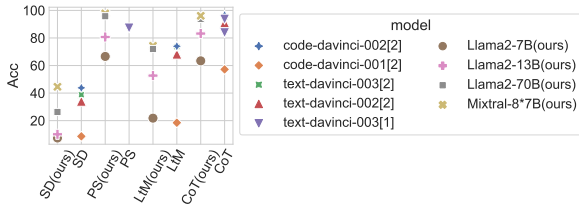
1170 Tab. 15 shows the results of self-consistency.



(a) GSM8K



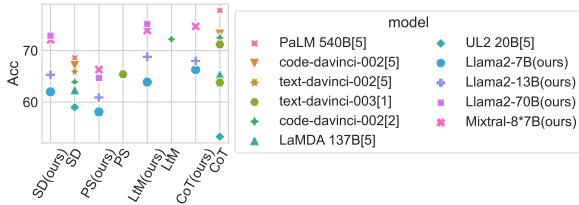
(b) ASDiv



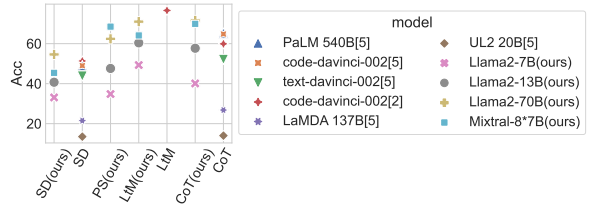
(c) MultiArith



(d) AQUA



(e) StrategyQA



(f) Date

Figure 7: A comparison of the results from existing work with the results reimplemented in this work for Llama2-7B, Llama2-13B, Llama2-70B, and Mixtral-7*8B across six datasets. The existing results come from five works: [1] (Wang et al., 2023a), [2] (Lyu et al., 2023), [3] (Luo et al., 2023a), [4] (Azerbaiyev et al., 2023), and [5] (Wei et al., 2022).

Case 1: Commonsense reasoning

Question: Does US brand Nice depend on Western honey bee?

True Answer: TRUE

CoT Response: Nice is a brand of ice cream. Ice cream is not made from honey bees. Thus, Nice does not depend on Western honey bees. So the answer is no. [WRONG]

CoT+HinSo Response:

Hint: Consider the crops that Nice brand depends on and whether those crops require pollination by Western honey bees.

Solution: Nice brand depends on almonds. Almonds require pollination by Western honey bees. Thus, Nice brand depends on Western honey bees. So the answer is yes. [CORRECT]

Case 2: Commonsense reasoning

Question: Do black-tailed jackrabbits fear the European wildcat?

True Answer: FALSE

CoT Response: Black-tailed jackrabbits are prey animals. European wildcats are predators. Thus, black-tailed jackrabbits would fear the European wildcat. So the answer is yes. [WRONG]

CoT+HinSo Response:

Hint: Consider the natural habitats of black-tailed jackrabbits and European wildcats and whether their paths would cross.

Solution: Black-tailed jackrabbits are native to North America. European wildcats are native to Europe. Thus, their paths would not naturally cross. So the answer is no. [CORRECT]

Case 3: Mathematical reasoning

Question: A point on the edge of a fan blade that is rotating in a plane 10 centimeters from the center of the fan. What is the distance traveled, in centimeters, by this point after 30 seconds when the fan runs at the rate of 300 revolutions per minutes? # Answer option: [A)750π, B)1500π, C) 1875π, D)3000π, E)7500π]

True Answer: D

CoT Response: The fan makes 300 revolutions in 60 seconds. So it makes $300/60 = 5$ revolutions in 1 second. So it makes $5*30 = 150$ revolutions in 30 seconds. The point on the edge of the fan blade travels 10 centimeters in one revolution. So it travels $10*150 = 1500$ centimeters in 30 seconds. So the answer is (b). [WRONG]

CoT+HinSo Response:

Hint: The distance traveled by a point on the edge of a rotating object is calculated by multiplying the circumference of the circle it moves along by the number of rotations.

Solution: The fan makes 300 revolutions in 60 seconds. In 30 seconds, it makes $300/2 = 150$ revolutions. The circumference of the circle is $2\pi r = 2\pi(10) = 20\pi$. The distance traveled is $150 * 20\pi = 3000\pi$. So the answer is (d). [CORRECT]

Case 4: Mathematical reasoning

Question: James wants to hang crepe-paper streamers from his backyard fence for his birthday party. His backyard fence is a rectangle that measures 20 feet on the long side and 15 feet on the short side. How many feet of crepe paper does James need to buy?

True Answer: 70

CoT Response: The fence is 20 feet long and 15 feet wide. The total length of the fence is $20 + 15 = 35$ feet. He needs exactly the same amount of crepe paper as the length of the fence. So he needs to buy 35 feet of crepe paper. The answer is 35. [WRONG]

CoT+HinSo Response:

Hint: To find out how much crepe paper James needs to buy, calculate the perimeter of the fence. The perimeter is the total distance around the fence. To find the perimeter, add the lengths of all four sides of the fence together.

Solution: The long side of the fence is 20 feet. The short side is 15 feet. So the perimeter is $20 + 15 + 20 + 15 = 70$ feet. The answer is 70. [CORRECT]

Table 14: Case studies of solving mathematical reasoning and commonsense reasoning problems with CoT+HinSo and CoT prompting on the Mixtral-7*8B model. Blue text indicates the stem, pink text indicates the effective hint, cyan text indicates the judgment of whether the answer is correct, [CORRECT] denotes correct, and [WRONG] denotes incorrect.

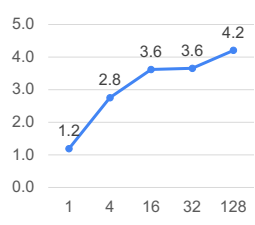
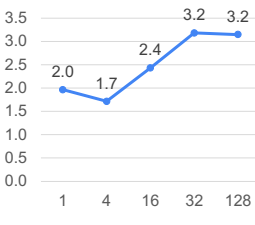
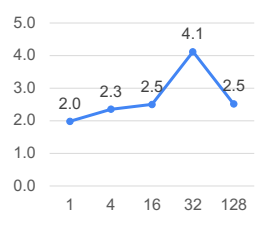
| Model | SC | Hint | MATH | | | | Commonsense | | | Avg | Relative Improvement |
|------------|------|------|-------------|-------------|-------------|-------------|-------------|-------------|------|---|----------------------|
| | | | GSM8K | ASDiv | MultiArith | AQUA | SQA | Date | | | |
| Llama2-7B | 1 | 0 | 19.7 | 53.6 | 63.4 | 24.4 | 66.3 | 40.1 | 44.6 |  | |
| | 1 | 1 | 19.9 | 55.8 | 63.8 | 24.4 | 67.5 | 43.2 | 45.8 | | |
| | 1 | Impv | 0.2 | 2.2 | 0.4 | 0.0 | 1.2 | 3.1 | 1.2 | | |
| | 4 | 0 | 23.6 | 54.6 | 68.0 | 23.6 | 67.9 | 40.1 | 46.3 | | |
| | 4 | 1 | 26.5 | 57.1 | 73.0 | 26.4 | 69.2 | 42.1 | 49.1 | | |
| | 4 | Impv | 2.9 | 2.5 | 5.0 | 2.8 | 1.3 | 2.0 | 2.8 | | |
| | 16 | 0 | 24.7 | 55.5 | 68.5 | 26.0 | 67.9 | 40.1 | 47.1 | | |
| | 16 | 1 | 29.2 | 57.3 | 77.7 | 26.0 | 70.7 | 43.5 | 50.7 | | |
| | 16 | Impv | 4.5 | 1.8 | 9.2 | 0.0 | 2.8 | 3.4 | 3.6 | | |
| | 32 | 0 | 25.5 | 55.2 | 67.6 | 25.6 | 68.6 | 39.6 | 47.0 | | |
| | 32 | 1 | 29.5 | 57.5 | 78.9 | 26.4 | 70.2 | 41.5 | 50.7 | | |
| | 32 | Impv | 4.0 | 2.3 | 11.3 | 0.8 | 1.6 | 1.9 | 3.6 | | |
| | 128 | 0 | 25.4 | 55.1 | 68.1 | 26.4 | 68.3 | 40.4 | 47.3 | | |
| | 128 | 1 | 30.3 | 59.0 | 79.5 | 25.6 | 70.2 | 44.3 | 51.5 | | |
| 128 | Impv | 4.9 | 3.9 | 11.4 | -0.8 | 1.9 | 3.9 | 4.2 | | | |
| Llama2-13B | 1 | 0 | 34.5 | 60.5 | 83.2 | 25.6 | 68.0 | 52.4 | 54.0 |  | |
| | 1 | 1 | 36.5 | 61.2 | 87.1 | 25.6 | 72.1 | 53.5 | 56.0 | | |
| | 1 | Impv | 2.0 | 0.7 | 3.9 | 0.0 | 4.1 | 1.1 | 2.0 | | |
| | 4 | 0 | 40.7 | 61.5 | 87.8 | 25.6 | 69.1 | 57.4 | 57.0 | | |
| | 4 | 1 | 41.1 | 62.7 | 89.4 | 28.7 | 72.6 | 57.9 | 58.7 | | |
| | 4 | Impv | 0.4 | 1.2 | 1.6 | 3.1 | 3.5 | 0.5 | 1.7 | | |
| | 16 | 0 | 42.3 | 62.5 | 89.4 | 28.0 | 69.0 | 57.7 | 58.2 | | |
| | 16 | 1 | 46.9 | 64.7 | 91.3 | 28.7 | 72.8 | 59.1 | 60.6 | | |
| | 16 | Impv | 4.6 | 2.2 | 1.9 | 0.7 | 3.8 | 1.4 | 2.4 | | |
| | 32 | 0 | 41.5 | 62.6 | 90.1 | 26.4 | 69.6 | 57.9 | 58.0 | | |
| | 32 | 1 | 48.2 | 64.9 | 92.8 | 28.0 | 73.1 | 60.2 | 61.2 | | |
| | 32 | Impv | 6.7 | 2.3 | 2.7 | 1.6 | 3.5 | 2.3 | 3.2 | | |
| | 128 | 0 | 47.9 | 62.9 | 90.1 | 27.6 | 70.2 | 58.8 | 59.6 | | |
| | 128 | 1 | 52.5 | 66.8 | 93.0 | 29.5 | 73.0 | 61.6 | 62.7 | | |
| 128 | Impv | 4.6 | 3.9 | 2.9 | 1.9 | 2.8 | 2.8 | 3.2 | | | |
| Llama2-70B | 1 | 0 | 46.1 | 72.5 | 93.8 | 35.8 | 74.6 | 71.6 | 65.7 |  | |
| | 1 | 1 | 50.3 | 74.4 | 94.6 | 37.0 | 77.0 | 73.0 | 67.7 | | |
| | 1 | Impv | 4.2 | 1.9 | 0.8 | 1.2 | 2.4 | 1.4 | 2.0 | | |
| | 4 | 0 | 59.5 | 75.0 | 95.3 | 39.8 | 78.2 | 73.3 | 70.2 | | |
| | 4 | 1 | 60.5 | 75.9 | 96.1 | 41.3 | 78.3 | 82.7 | 72.5 | | |
| | 4 | Impv | 1.0 | 0.9 | 0.8 | 1.5 | 0.1 | 9.4 | 2.3 | | |
| | 16 | 0 | 60.1 | 76.3 | 96.1 | 42.1 | 78.4 | 73.3 | 71.1 | | |
| | 16 | 1 | 67.0 | 77.9 | 97.8 | 44.5 | 79.2 | 74.9 | 73.6 | | |
| | 16 | Impv | 6.9 | 1.6 | 1.7 | 2.4 | 0.8 | 1.6 | 2.5 | | |
| | 32 | 0 | 60.6 | 77.1 | 96.3 | 45.3 | 78.5 | 72.7 | 71.8 | | |
| | 32 | 1 | 67.4 | 78.4 | 98.0 | 47.2 | 80.1 | 84.1 | 75.9 | | |
| | 32 | Impv | 6.8 | 1.3 | 1.7 | 1.9 | 1.6 | 11.4 | 4.1 | | |
| | 128 | 0 | 67.0 | 77.6 | 96.3 | 46.1 | 79.4 | 73.5 | 73.3 | | |
| | 128 | 1 | 67.6 | 78.8 | 98.2 | 47.6 | 79.5 | 83.3 | 75.8 | | |
| 128 | Impv | 0.6 | 1.2 | 1.9 | 1.5 | 0.1 | 9.8 | 2.5 | | | |

Table 15: The results of self-consistency on the six datasets. Values in green denote the relative performance improvement with hints versus without hints under the same setting. The blue bold values represent the best performance with hints, while the pink bold values indicate the best performance without hints. The figure on the right shows the average relative improvement across six datasets.