Enhancing Chain of Thought Prompting in Large Language Models via Reasoning Patterns

Yufeng Zhang[1,2], **Xuepeng Wang**[1,3,*], **Lingxiang Wu**[1,3,*], **Jinqiao Wang**[1,2,3]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]Wuhan AI Research, Wuhan, China
{yufeng.zhang, xuepeng.wang}@ia.ac.cn, {lingxiang.wu, jqwang}@nlpr.ia.ac.cn

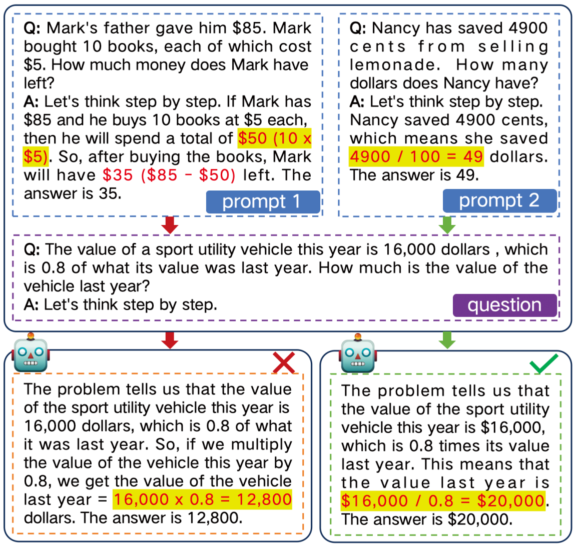
Abstract

Chain of Thought (CoT) prompting can encourage language models to engage in multi-step logical reasoning. The quality of the provided demonstrations significantly influences the success of downstream inference tasks. Current unsupervised CoT methods primarily select examples based on the semantics of the questions, which can introduce noise and lack interpretability. In this paper, we propose leveraging reasoning patterns to enhance CoT prompting effectiveness. Reasoning patterns represent the process by which language models arrive at their final results. By utilizing prior knowledge and prompt-based methods from large models, we first construct task-specific pattern sets. We then select diverse demonstrations based on different reasoning patterns. This approach not only mitigates the impact of noise but also provides explicit interpretability to help us understand the mechanisms of CoT. Extensive experiments demonstrate that our method is more robust and consistently leads to improvements across various reasoning tasks.

Introduction

Large Language Models (LLMs) have demonstrated exceptional performance across a wide range of language tasks. In general question-answering tasks (Kwiatkowski et al. 2019), LLMs hold a distinct advantage over other language models due to their robust writing capabilities. However, when it comes to more advanced tasks such as logical reasoning, mathematical computation, and symbolic reasoning, LLMs often fall short (Qiao et al. 2023; Huang and Chang 2023).

One effective approach to addressing these challenges is Chain of Thought (CoT) prompting (Wei et al. 2022b). By providing several demonstration examples that include a problem, intermediate reasoning steps, and an answer, CoT prompting serves as a contextual guide for downstream tasks. This approach encourages LLMs to generate multi-step logical reasoning, thereby maximizing the likelihood of producing more plausible answers. The advantage of this method lies in its simplicity and efficiency; unlike fine-tuning, it does not require extensive gradient updates or alter the model's inherent capabilities. Instead, it acts as an external augmentation of knowledge. For different reasoning tasks, we can route the model to the appropriate context, and then easily switch the demonstration sets to activate the relevant knowledge and abilities in the corresponding domain.



Q: Mark's father gave him $85. Mark bought 10 books, each of which cost $5. How much money does Mark have left?
A: Let's think step by step. If Mark has $85 and he buys 10 books at $5 each, then he will spend a total of $50 (10 x $5). So, after buying the books, Mark will have $35 ($85 - $50) left. The answer is 35.

Q: Nancy has saved 4900 cents from selling lemonade. How many dollars does Nancy have?
A: Let's think step by step. Nancy saved 4900 cents, which means she saved 4900 / 100 = 49 dollars. The answer is 49.

Q: The value of a sport utility vehicle this year is 16,000 dollars , which is 0.8 of what its value was last year. How much is the value of the vehicle last year?
A: Let's think step by step.

The problem tells us that the value of the sport utility vehicle this year is 16,000 dollars, which is 0.8 of what it was last year. So, if we multiply the value of the vehicle this year by 0.8, we get the value of the vehicle last year = 16,000 x 0.8 = 12,800 dollars. The answer is 12,800.

The problem tells us that the value of the sport utility vehicle this year is $16,000, which is 0.8 times its value last year. This means that the value last year is $16,000 / 0.8 = $20,000. The answer is $20,000.

Figure 1: Example of the chain-of-thought prompting. The prompt influences how LLMs arrive at the final answer.

However, we argue that existing unsupervised CoT prompting methods have two major shortcomings. First, there remains a significant gap between the selected demonstration sets and the reasoning targets. Although extensive research (Zhang et al. 2023; Levy, Bogin, and Berant 2023; Yang et al. 2023; Shum, Diao, and Zhang 2023a) has explored ways to provide CoT demonstrations to enhance LLMs' reasoning capabilities, these methods largely rely on the semantic features of the problem or the answer. Such features introduce irrelevant noise on a global scale, which can obscure the logical information needed for reasoning. Consequently, the constructed demonstration sets do not effectively represent the domain-specific logical knowledge, and struggle to adequately trigger correct reasoning in LLMs. Second, some demonstration selection methods lack interpretability and scalability. These methods are primarily based on heuristic design (Wang et al. 2022; Zheng

[*] Corresponding authors.

et al. 2023) or leverage the model itself to generate additional demonstrations (Zhong et al. 2024; Yasunaga et al. 2024). The demonstration sets chosen through these means inherently lack clear explanations, making it challenging to assess their effectiveness or determine the direction for further optimization. This limitation can be particularly problematic in scenarios where interpretability is crucial.

To better select a demonstration subset for a reasoning task, we believe that considering the logical patterns of reasoning is essential. Inspired by the work of (Min et al. 2022) and (Madaan, Hermann, and Yazdanbakhsh 2023), we observe that LLMs are more influenced by the templates and patterns in the context than by the correctness of the demonstrations themselves. Building on this insight, we investigate the selection of demonstrations based on **Reasoning Patterns**. This approach offers a dual benefit. First, it helps to eliminate bias introduced by irrelevant information, thereby reducing the gap between the demonstration set and the reasoning task. Second, it provides explicit interpretability, allowing us to gain a deeper understanding of how CoT prompting functions. This interpretability can also serve as a clue for attribution analysis and visualization.

In this work, we propose **Pattern-CoT**[1], a CoT demonstration selection method based on reasoning patterns. Unlike previous approaches that focus on overall semantics, our method targets finer-grained logical reasoning operations. For instance, in mathematical reasoning, addition and multiplication represent distinct operations, while multiple sequential operators may indicate more complex operational patterns, as shown in Figure 1. Inspired by recent studies (Yang et al. 2023), a diverse range of these patterns should be incorporated into CoT. Specifically, for a given reasoning task, we first obtain a set of seed demonstrations with rationale (intermediate reasoning steps). These examples can be sourced from the training set or generated using a zero-shot approach. We then obtain specific operation tokens tailored to different task types, which help us extract reasoning patterns from the rationales. Here, we incorporate prior knowledge and guide the LLMs in generating these operation tokens. Based on the extracted reasoning patterns, we apply clustering techniques to merge similar patterns and design metrics to automatically assess the number of demonstration categories. Finally, we select representative demonstrations from each category to enrich the diversity and construct context prompts for LLMs. Notably, by incorporating task-specific knowledge, our method improves interpretability and facilitates further scalability.

Our contributions can be summarized as follows:

- We introduce the use of diverse reasoning patterns to enhance CoT prompting and design a demonstration selection method to reduce the gap between the demonstration set and the task.
- Our method strengthens the interpretability of CoT in unsupervised scenarios, and can be utilized for further attribution analysis.
- Extensive experiments demonstrate that our method consistently enhances performance across multiple reasoning tasks and various models.

Related Work

Chain-of-Thought Prompting

Large language models have demonstrated significant ability in comprehending context and responding to prompts (Brown et al. 2020; Ouyang et al. 2022). Recent studies highlight that LLMs can achieve improved task completion without fine-tuning, particularly on reasoning tasks, when provided with few-shot demonstrations (Wei et al. 2022b). For instance, when presented with an example like *Q: Mary has 9 yellow marbles. John has 3 yellow marbles. How many yellow marbles do they have in all? A: They have 9 + 3 = 12 yellow marbles. The answer is 12*, LLMs are expected to emulate such a format, deconstruct the question, engage in multi-step reasoning, and refrain from generating random answers in subsequent tasks. This process is commonly referred to as chain-of-thought prompting or in-context learning (Wei et al. 2022a; Xie et al. 2022). However, implementing this practice often involves the manual design of prompts at a labour cost. Consequently, researchers are exploring more efficient example selection strategies to streamline this process.

Demonstration Selection and Refinement

Several CoT studies are directed towards automating the generation of demonstrations, such as retrieval-based (Rubin, Herzig, and Berant 2022), zero-shot (Kojima et al. 2022), clustering-based (Zhang et al. 2023), and self-prompt (Shao et al. 2023; Yasunaga et al. 2024). However, many of these approaches encounter challenges in achieving performance comparable to Manual-CoT, primarily due to the absence of supervision in example selection. In another branch of research, efforts are focused on enhancing the quality of CoT demonstrations. They incorporate elements such as knowledge-infusion (Zhao et al. 2023; Weng et al. 2023; Li et al. 2024), self-consistency (Wang et al. 2023a), complexity-based (Fu et al. 2022), contrastive-based (Chia et al. 2023), and progressive-hint (Zheng et al. 2023). The primary goal of these strategies is to ensure that LLMs adhere to the correct prompt and avoid being misled.

Role of In-Context Patterns

To understand the underlying mechanism of ICL, (Min et al. 2022) and (Madaan, Hermann, and Yazdanbakhsh 2023) employ counterfactual prompting methods. These methods involve substituting question-answer mapping, token distributions, answer patterns, and many other factors. Their findings consistently show that the correctness of examples is not the most crucial factor, but rather the distribution or pattern (e.g. equations, templates, sentence structure) of the examples. In this paper, we continue to uncover the power of CoT patterns and show how they can improve the reasoning process.

[1] https://github.com/Magicat128/Pattern-CoT.

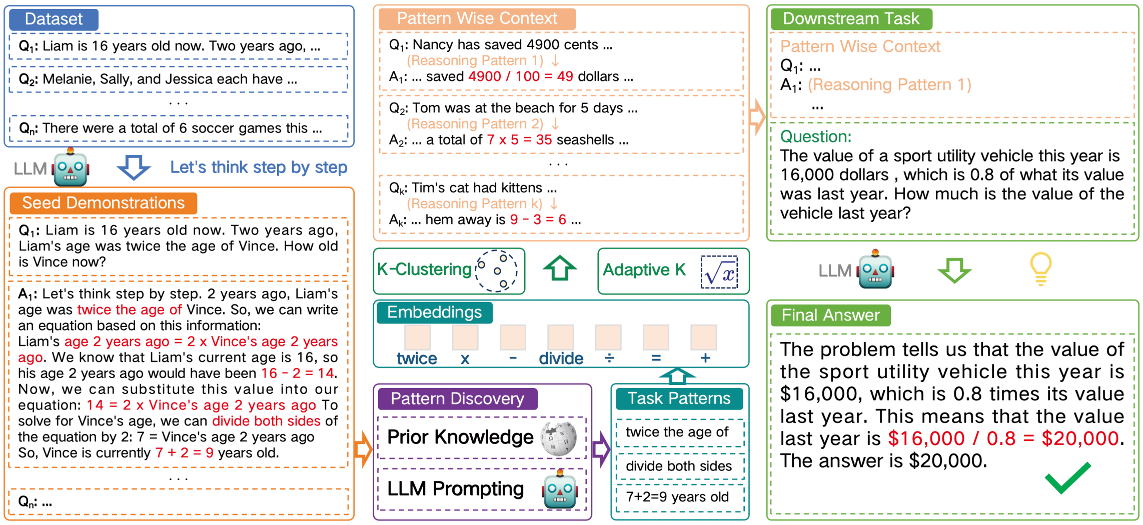

Dataset
Q1: Liam is 16 years old now. Two years ago, ...
Q2: Melanie, Sally, and Jessica each have ...
Qn: There were a total of 6 soccer games this ...
LLM
Let's think step by step
Seed Demonstrations
Q1: Liam is 16 years old now. Two years ago, Liam's age was twice the age of Vince. How old is Vince now?
A1: Let's think step by step. 2 years ago, Liam's age was twice the age of Vince. So, we can write an equation based on this information:
Liam's age 2 years ago = 2 x Vince's age 2 years ago. We know that Liam's current age is 16, so his age 2 years ago would have been 16 - 2 = 14.
Now, we can substitute this value into our equation: 14 = 2 x Vince's age 2 years ago To solve for Vince's age, we can divide both sides of the equation by 2: 7 = Vince's age 2 years ago
So, Vince is currently 7 + 2 = 9 years old.
Qn: ...
Pattern Wise Context
Q1: Nancy has saved 4900 cents ...
(Reasoning Pattern 1) ↓
A1: ... saved 4900 / 100 = 49 dollars ...
Q2: Tom was at the beach for 5 days ...
(Reasoning Pattern 2) ↓
A2: ... a total of 7 x 5 = 35 seashells ...
Qk: Tim's cat had kittens ...
(Reasoning Pattern k) ↓
Ak: ... hem away is 9 - 3 = 6 ...
K-Clustering
Adaptive K
Embeddings
twice
x
-
divide
÷
=
+
Pattern Discovery
Prior Knowledge
LLM Prompting
Task Patterns
twice the age of
divide both sides
7+2=9 years old
Downstream Task
Pattern Wise Context
Q1: ...
A1: (Reasoning Pattern 1)
...
Question:
The value of a sport utility vehicle this year is 16,000 dollars , which is 0.8 of what its value was last year. How much is the value of the vehicle last year?
LLM
Final Answer
The problem tells us that the value of the sport utility vehicle this year is $16,000, which is 0.8 times its value last year. This means that the value last year is $16,000 / 0.8 = $20,000.
The answer is $20,000.
✓

Figure 2: Illustration of our proposed framework. We first extract different patterns from the original rationales. Then clustering is used to produce a group of demonstrations. This enables LLMs to perceive diverse reasoning patterns and to select a proper solution path. It avoids LLMs being biased by monotonous reasoning mode.

Methodology

We now explore the impact of diverse demonstration reasoning patterns on chain-of-thought prompting. According to (Min et al. 2022), the precision of demonstrations is not crucial when LLMs engage in ICL. Even if all the demonstrations provided are incorrect, it would only marginally impede performance. This aligns with the insight derived from Auto-CoT (Zhang et al. 2023): clustering zero-shot question-answer pairs without emphasizing accuracy can still yield valuable examples. Consequently, our focus shifts to a more nuanced factor - the underlying reasoning pattern that harbours more informative content (Madaan, Hermann, and Yazdanbakhsh 2023) - to evaluate its potential benefits for the CoT process. The entire process is summarized in Figure 2 and Algorithm 1.

Seed Demonstration Collection

For a given task $Q = \{q_1, q_2, ..., q_N\}$ with N questions, we first need to obtain their rationales and answers $\{q_i, r_i, a_i\}$ that can be used as context for CoT prompting. For data from existing training sets, we can directly use the training data. However, in practical applications, complete training sets may not always be available. In such cases, we refer to methods like (Zhang et al. 2023; Shum, Diao, and Zhang 2023b) and leverage the zero-shot (Kojima et al. 2022) capabilities of LLMs to generate the corresponding rationales. It is important to note that we do not require the answers to be correct or labelled; our focus is on whether the generated rationales contain meaningful reasoning patterns.

Pattern Discovery

Based on the rationale set $Ra = \{r_1, r_2, ..., r_N\}$ that we have obtained, we next identify the reasoning operations T associated with the task. For tasks with a relatively limited action space, we can define reasoning operations using prior knowledge, as these operations represent the fundamental units of reasoning tasks. For example, in arithmetic problems, we refer to a glossary of possible operators from sources like Wikipedia[2], including basic arithmetic operations, square roots, comparison symbols, etc. For tasks with less clearly defined operations, we adapt definitions from arithmetic problems to guide LLMs in generating the corresponding reasoning operations. We design the prompt as: *'Similar to operators used in arithmetic such as (+, -, *, /), which operators do you think best represent the [TASK]? Example of [TASK]: ...'*

For each rationale $r_i \in Ra$, we extract the reasoning operation tokens or phrases $t_j \in T$ to form its reasoning pattern:

$$p_i = f(r_i, T) = \{t_{i1}, t_{i2}, ..., t_{ij}\} \quad (1)$$

where f denotes the function used to extract the reasoning path. In this context, p_i represents how LLMs apply these operations step-by-step to reach the final result, and t_{ij} can repeated.

Pattern Wise Demonstration Selection

Once we have identified the task-relevant patterns, we use them to select better demonstration sets. Following (Zhang et al. 2023), we cluster all the p_i patterns while preserving diversity. Although p_i is a simplified sequence of tokens, it still contains substantial semantic information that can be used to uncover underlying similarities. For instance, a sequence of addition operations is likely to be closer to a single addition operation than to a single multiplication operation. To leverage this, we use a language model to encode these patterns. We then apply the k-means clustering algorithm to generate k clusters and sample from each cluster:

$$\widetilde{p_i} = \text{encode}(p_i) \quad (2)$$

[2] The glossary of arithmetic operators refers to the Wikipedia: https://en.wikipedia.org/wiki/Glossary_of_mathematical_symbols

Algorithm 1: Pattern-CoT Demonstration Selection

```
Require: A set of task questions $Q$
Ensure: Demonstration list $d = [d_1, d_2, ..., d_k]$
Acquire operation token set $T$ with LLMs prompting or domain knowledge based on $Q$
for $q_i \in Q$ do
    Generate rationale $r_i$ with Zero-Shot-CoT
    $p_i = []$
    for each token $t_{ij} \in r_i$ do
        if $t_{ij} \in T$ then
            Update $p_i$ with $t_{ij}$
        end if
    end for
    $\widetilde{p}_i = \text{encode}(p_i)$
end for
Select proper $k$
Cluster all $[\widetilde{p}_1, \widetilde{p}_2, ..., \widetilde{p}_i]$ into $k$ clusters
Sample $d = [d_1, d_2, ..., d_k]$ from each cluster
return $d$
```

$$c_1, c_2, ..., c_k = \text{cluster}(\widetilde{p}_1, \widetilde{p}_2, ..., \widetilde{p}_i) \tag{3}$$

$$d = \{q_m, r_m, a_m | \widetilde{p}_m \in c_m, m = 1, 2, ..., k\} \tag{4}$$

where d denotes the demonstration set, c_k denotes the k-th cluster. Specifically, we use patterns primarily to select demonstrations rather than directly as context for downstream tasks. We utilize the original problem q_k and rationale r_k corresponding to the p_k patterns as the CoT input.

Number of Demonstrations

Since previous methods lack knowledge-based guidance, the choice of k is often based on heuristic values. However, having too many demonstrations does not proportionally enhance the performance (Wei et al. 2022b; Agarwal et al. 2024), while too few may fail to adequately capture the task's characteristics. By incorporating reasoning operations, we can use the number of these operations to inform a more reasonable choice for k:

$$k = \lceil \frac{1}{2} \times n \times (1 + \log(N)) \rceil \tag{5}$$

where n denotes the number of identified operations, and $\lceil\rceil$ represents the ceiling function that rounds up to the nearest integer. This formula empirically takes into account the impact of the number of operation types on the number of demonstrations and further adjusts based on the sample size.

Experiments

In this section, our objective is to evaluate the effectiveness of our proposed method and answer the following research questions:

- **RQ1**: Does incorporating reasoning patterns enhance the effectiveness of CoT prompting?
- **RQ2**: How do the reasoning patterns influence the outputs of LLMs?
- **RQ3**: Is our method robust and scalable to other models?

Dataset	Samples	Operation Tokens
GSM8K	1319	$+, -, \times, /$ 'more', 'less', 'twice', 'half'
AQuA	254	$+, -, \times, /, \pi, \sqrt{x}, x^n, x^\circ, log$
MultiArith	600	$+, -, \times, /$
AddSub	395	
SingleEq	508	
SVAMP	1000	
Coin	500	'heads up', 'tails up'
Date	369	'day', 'week', 'month', 'year' 'yesterday', 'tomorrow'

Table 1: The number of samples and operation tokens.

Experimental Setup

Datasets. We adopt eight representative datasets for our reasoning tasks: MultiArith (Roy and Roth 2015), GSM8K (Cobbe et al. 2021), AddSub (Hosseini et al. 2014), AQUA-RAT (Ling et al. 2017), SingleEq (Koncel-Kedziorski et al. 2015), SVAMP (Patel, Bhattamishra, and Goyal 2021), Coin-Flip (Wei et al. 2022b), and BIG-bench Date Understanding (Srivastava et al. 2023). They require certain reasoning steps and are commonly used for CoT method comparisons (Wei et al. 2022b; Kojima et al. 2022; Zhang et al. 2023; Wang et al. 2023b; Fu et al. 2022).

For tasks MultiArith, AddSub, SingleEq, and SVAMP, we define the set of operation tokens based on a glossary from Wikipedia, as the operations involved are relatively straightforward. For tasks GSM8K and AQUA, we expand the operation token vocabulary manually based on data distribution. For tasks Coin-Flip and BIG-bench Date Understanding, we prompt GPT-4 to generate the corresponding operation tokens. The specific details of the datasets can be found in Table 1.

Language Models. To facilitate subsequent interpretability analysis, we select open-source models as our reasoning engine. Specifically, we use models from the LLaMA-2 family due to their foundational logical reasoning capabilities and support for CoT prompting. These models are deployed on our local server, which is equipped with 8 RTX 3090 GPUs, each with 24GB of memory. Due to hardware constraints, we test only the 7B and 13B models. Experiments with larger models or those from other families are discussed in subsequent sections.

We use the inference functions of these models, and the process does not involve training or fine-tuning. Additionally, we set the hyperparameters with a temperature of 0.4 and top_p of 0.9 to manage the model's randomness (Xu et al. 2022). To maintain consistency with (Zhang et al. 2023), we use Sentence-BERT (Reimers and Gurevych 2019) as our encoder and select the *'all-MiniLM-L6-v2'* model for semantic vector representation. This model has also been proven effective in our experiments.

LLaMA-2 Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	Coin	Date
7b-chat-hf	Zero-Shot-CoT	72.33	21.00	57.97	24.01	57.67	41.90	44.60	39.29
	(+ SC)	**79.83**	27.14	62.78	21.65	68.11	47.60	52.80	40.37
	Random-CoT	76.16	24.41	65.59	22.44	66.14	46.59	48.00	44.44
	Auto-CoT	76.00	26.99	58.48	24.01	64.96	43.80	51.20	44.71
	Auto-CoT-RA	74.83	26.76	63.29	23.80	66.92	45.19	48.00	43.08
	Ours	79.66	27.45	65.06	28.34	**71.85**	**48.50**	**59.40**	45.79
	Ours (Adaptive k)	79.66*	**28.05**	**67.08**	**29.13**	71.85*	48.50*	58.40	**46.34**
13b-chat-hf	Zero-Shot-CoT	77.50	34.49	60.75	15.74	69.29	49.40	47.40	46.07
	Auto-CoT	82.16	36.77	63.03	25.19	70.67	55.50	54.20	53.93
	Auto-CoT-RA	82.16	37.04	62.08	27.74	66.14	52.10	62.80	54.47
	Ours	**83.16**	37.68	**65.82**	26.37	**74.80**	**56.39**	57.40	56.91
	Ours (Adaptive k)	83.16*	**38.44**	64.81	**31.49**	74.80*	56.39*	**67.80**	**60.97**

Table 2: Accuracy (%) on eight reasoning datasets. We present the mean value obtained from five runs. * denotes the situation where k does not change, and results are copied from above. For the Random-CoT method, we report the best result since we are concerned about the potential of CoT. For the self-consistency method, we set the number of paths as 5 (Wang et al. 2023a).

Baselines. We primarily compare our methods with unsupervised methods including Zero-Shot-CoT (Kojima et al. 2022), Random-CoT, Auto-CoT (Zhang et al. 2023), and Self-Consistency (Wang et al. 2023a). Building on Auto-CoT, we introduce an additional variant, Auto-CoT-RA, which replaces the original question embedding with the rationale embedding for clustering. The purpose of this modification is to investigate whether this subtle shift can implicitly uncover the underlying patterns in reasoning. Unless otherwise specified, our method uses the same k value as the baseline in experiments. Additionally, we conduct experiments using our method with the adaptive k value that we designed.

Main Results (RQ1)

Table 2 presents the overall performance of various methods on the 7B and 13B models. Since our primary goal is to evaluate whether focusing on diverse patterns provides more benefit to reasoning than semantic information, we are not concerned with identifying which model achieves state-of-the-art performance. Based on these results, we make the following observations:

- Overall, our method consistently outperforms the baseline approaches. This stable improvement indicates that by introducing diverse reasoning patterns, we can identify more representative demonstration sets, where each example embodies a different reasoning strategy. Using these diverse examples as context for LLMs can further enhance their ability to solve downstream tasks.
- We observe that for arithmetic problems with a limited set of operation tokens, such as MultiArith, AddSub, SingleEq, and SVAMP, our method achieves more significant improvements compared to methods based on semantic information. This suggests that the demonstration sets we construct can effectively cover the majority of reasoning paths, thereby providing comprehensive guidance for LLMs to select appropriate reasoning patterns.

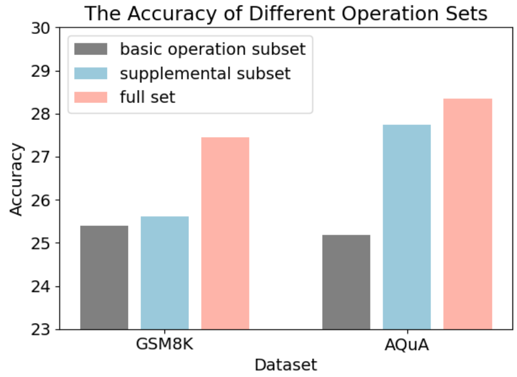


Figure 3: Comparison of different operation sets.

- For datasets with a relatively broader action space, like GSM8K and AQuA, the improvements are less significant. This implies that a limited number of examples do not fully capture the diversity of reasoning patterns. However, when we recalculate the number of clusters using adaptive k and expand the demonstration set, we observe additional gains on these two datasets.
- Surprisingly, we find that for datasets like Coin and Date, where the operation patterns are not explicitly defined, our method actually lead to greater improvements. We hypothesize that this is because the questions in these datasets are quite similar, making it difficult to distinguish them based on semantic features alone. In contrast, leveraging reasoning patterns allows us to identify representative examples from a different dimension. Additionally, both of these datasets show further performance improvements when using adaptive k.

There are several additional observations. For instance, in some cases, Auto-CoT-RA outperforms Auto-CoT, while in others it does not. This suggests that simply shifting from

Auto-CoT	**Ours**
Q: Mark's father gave him $85. Mark bought 10 books, each of which cost $5. How much money does Mark have left? **A:** Let's think step by step. If Mark has $85 and he buys 10 books at $5 each, then **he will spend a total of $50 (10 x $5)**. So, after buying the books, **Mark will have $35 ($85 - $50) left**. The answer is 35. **Q:** Mike has 48 books. Alyssa has 8 times more books than Mike. How many dozen books does Alyssa have? **A:** Let's think step by step. If Mike has 48 books, then Alyssa has 8 **times that number = 8 x 48 = 384** books. So Alyssa has 384 books. The answer is 384.	**Q:** Tom was at the beach for 5 days and found 7 seashells every day. How many seashells did Tom find during the beach trip? **A:** Let's think step by step. If Tom found 7 seashells every day, then he found a total of **7 x 5 = 35** seashells during his entire beach trip. The answer is 35. **Q:** Nancy has saved 4900 cents from selling lemonade. How many dollars does Nancy have? **A:** Let's think step by step. Nancy saved 4900 cents, which means she saved **4900 / 100 = 49** dollars. The answer is 49.
Q: The value of a sport utility vehicle this year is 16,000 dollars , which is 0.8 of what its value was last year. How much is the value of the vehicle last year? **A:** Let's think step by step. The problem tells us that the value of the sport utility vehicle this year is 16,000 dollars, which is 0.8 of what it was last year. So, **if we multiply the value of the vehicle this year by 0.8, we get the value of the vehicle last year = 16,000 x 0.8 = 12,800 dollars**. The answer is 12,800.	**Q:** The value of a sport utility vehicle this year is 16,000 dollars , which is 0.8 of what its value was last year. How much is the value of the vehicle last year? **A:** Let's think step by step. The problem tells us that the value of the sport utility vehicle this year is $16,000, which is 0.8 times its value last year. This means that **the value last year is $16,000 / 0.8 = $20,000**. The answer is $20,000.

Table 3: Case study of Auto-CoT and our method for demonstration selection and downstream inference. The upper questions and answers are demonstrations constructed by two methods, and the lower part contains how LLMs solve the inference task.

Dataset	Demos	Incorrect	Error Rate
MultiArith	8	2	25.0%
GSM8K	8	5	62.5%
AddSub	8	3	37.5%
AQuA	4	4	100%
SingleEq	8	2	25.0%
SVAMP	8	6	75%
Coin	8	3	37.5%
Date	8	1	12.5%

Table 4: The number of demonstrations and their error rate for each dataset.

question semantics to rationale semantics does not necessarily narrow the gap between demonstrations and the reasoning task. Deeper reasoning patterns can still be obscured by irrelevant information. Moreover, in certain situations, using a random demonstration set can also surpass Auto-CoT, although this improvement is inconsistent. This indirectly highlights that other factors, such as underlying reasoning patterns, can influence the effectiveness of examples. Our method, in most cases, demonstrates a more stable ability to uncover these factors.

Impact of Operation Tokens (RQ1)

To further assess the impact of reasoning patterns, we conduct additional experiments. Given that GSM8K and AQuA datasets utilize additional operation tokens, we removed some of these tokens to determine their influence. Specifically, we categorize the expanded operation tokens into a basic operation subset, such as $\{+, -, \times, /\}$, similar to other arithmetic tasks, and the remaining tokens as supplementary subsets. These subsets represent only a portion of the reasoning patterns within these two datasets.

Figure 3 shows the results of using different subsets on the 7B model. The experimental results demonstrate that using operation subsets as reasoning pattern tokens can degrade overall performance. The primary reason for this is that these subsets do not sufficiently cover the task's logical scope. It leads to a lack of diversity. However, when the full set of operations is utilized, a broader range of scenarios can be activated, allowing the model to better adapt to the task.

Case Study (RQ2)

To gain a deeper understanding of CoT prompting, we perform a case study. Table 3 presents a typical instance analysis. We observe that Auto-CoT, due to its introduction of numerous irrelevant patterns, tends to distort the reasoning results of LLMs. In contrast, our method, which includes a diverse set of reasoning pattern templates, enables the model to generate correct responses.

Feature Attribution (RQ2)

Following the previous case study, we seek to understand why different contextual reasoning patterns alter the output of LLMs. Specifically, we employ a perturbation-based feature attribution analysis method (Winter 2002) to aid in this understanding. Traditional attention-based analysis methods have been criticized for their inability to identify the most significant features (Wiegreffe and Pinter 2019; Zhao et al. 2024), which is why we turned to this perturbation-based approach. By masking portions of the input tokens, we recompute the generation probabilities for each output token to as-

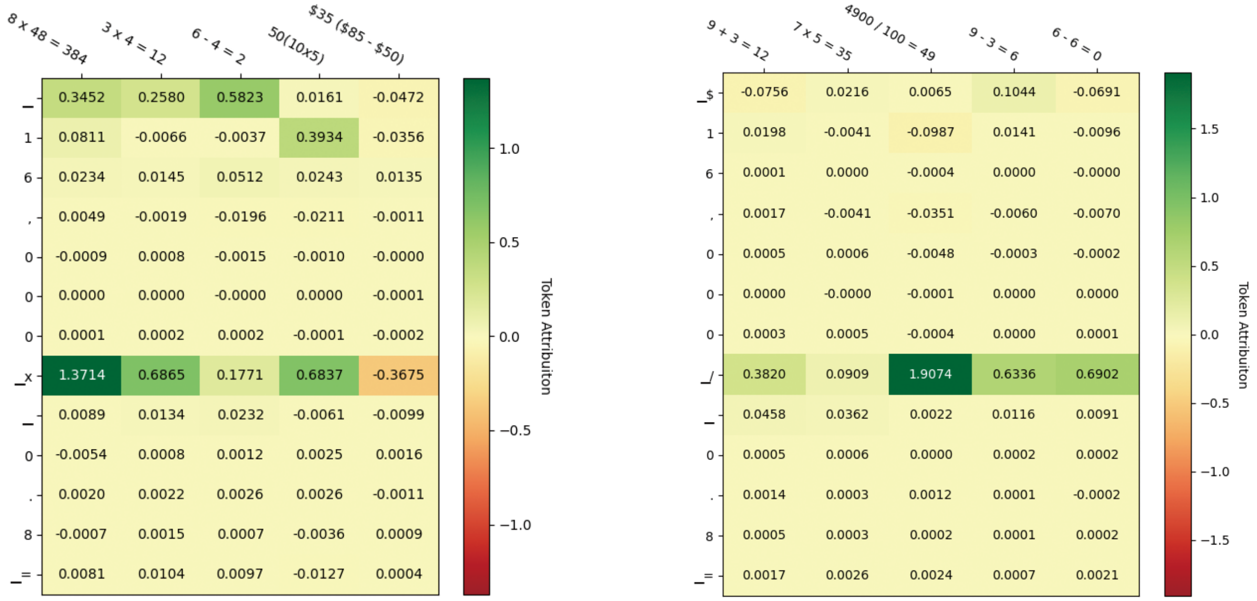


Figure 4: Visualization of token attribution for the case study. The left part stands for the score matrix of patterns from Auto-CoT, and the right part stands for the score matrix from our method. The upper column denotes each individual prompt, and the row denotes the generated token sequence. Higher scores (positive) indicate that the input has a greater impact on the output.

Model		AddSub	AQuA	SingleEq
GPT-3.5	Zero-Shot	83.29	59.44	90.55
	Auto-CoT	81.26	58.66	91.53
	Ours	**83.54**	**62.38**	**93.11**
Qwen	Zero-Shot	54.93	**35.03**	69.07
	Auto-CoT	62.53	30.31	80.31
	Ours	**67.59**	33.46	**82.08**

Table 5: Result of GPT-3.5-turbo-0125 and Qwen-7b-chat model on different datasets.

sess the input's attribution impact on these output tokens. We use Captum (Miglani et al. 2023) to achieve this visualization. Figure 4 presents the attribution analysis matrix for the case study. According to the visualization results, we find that when a particular pattern is overly dense in the examples, the model tends to activate related knowledge, which can lead to biased reasoning processes. Conversely, when these patterns are more diverse, the model is more likely to activate the correct reasoning pathways. Our method, by enhancing the diversity of patterns in the demonstrations, effectively reduces the distance to the reasoning task objectives.

Error Robustness (RQ3)

It is worth mentioning that we do not impose supervision on the labels of the demonstrations. Therefore, we proceed to count the number of incorrect instances within the selected set, as shown in Table 4. It is intriguing to notice that the majority of our provided demonstrations are imperfect, with AQuA even exhibiting a 100% error rate. This phenomenon suggests that LLMs struggle to discern incorrect instances from correct ones. Instead, they learn from how the example approaches problem-solving, which we refer to as 'pattern'. Our method encourages LLMs to follow the most probable reasoning chain towards the final answer and thus leads to a significant improvement.

Results on Other Models (RQ3)

To determine whether our method is applicable to different models, we test it on various LLM branches. Specifically, we select the GPT series to represent larger and more advanced models, and Qwen to represent multilingual models. For the sake of hardware resources and budget constraints, we experiment with the GPT-3.5-turbo and Qwen-7B models. Table 5 presents the performance of several methods on these models. Notably, the experiments show that Auto-CoT, in some cases, underperforms compared to direct answering on these models. We attribute this to the inherent noise in semantics-based methods. Our approach mitigates this noise, resulting in more consistent performance improvements.

Conclusion

This paper aims to address the noise issue inherent in unsupervised semantic-based CoT methods and proposes a reasoning pattern-based approach for CoT demonstration selection. Our method explicitly enhances the interpretability of reasoning processes and illustrates how LLMs can be guided toward generating accurate answers. Extensive experiments validate the effectiveness, robustness, and compatibility of our approach.

Acknowledgements

This work was supported by the National Key R&D Program of China (Grant No.2023ZD0120400), Beijing Natural Science Foundation (L247028), National Natural Science Foundation of China (No. 62276260, 62076235), Beijing Municipal Science and Technology Project (Z231100007423004). We sincerely thank all reviewers and ACs for their insightful comments, time and efforts.

References

Agarwal, R.; Singh, A.; Zhang, L. M.; Bohnet, B.; Rosias, L.; Chan, S.; Zhang, B.; Anand, A.; Abbas, Z.; Nova, A.; Co-Reyes, J. D.; Chu, E.; Behbahani, F.; Faust, A.; and Larochelle, H. 2024. Many-Shot In-Context Learning. arXiv:2404.11018.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901.

Chia, Y. K.; Chen, G.; Tuan, L. A.; Poria, S.; and Bing, L. 2023. Contrastive Chain-of-Thought Prompting. arXiv:2311.09277.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Fu, Y.; Peng, H.; Sabharwal, A.; Clark, P.; and Khot, T. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Hosseini, M. J.; Hajishirzi, H.; Etzioni, O.; and Kushman, N. 2014. Learning to Solve Arithmetic Word Problems with Verb Categorization. In Moschitti, A.; Pang, B.; and Daelemans, W., eds., *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 523–533. Doha, Qatar: Association for Computational Linguistics.

Huang, J.; and Chang, K. C.-C. 2023. Towards Reasoning in Large Language Models: A Survey. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 1049–1065. Toronto, Canada: Association for Computational Linguistics.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213.

Koncel-Kedziorski, R.; Hajishirzi, H.; Sabharwal, A.; Etzioni, O.; and Ang, S. D. 2015. Parsing Algebraic Word Problems into Equations. *Transactions of the Association for Computational Linguistics*, 3: 585–597.

Kwiatkowski, T.; Palomaki, J.; Redfield, O.; Collins, M.; Parikh, A.; Alberti, C.; Epstein, D.; Polosukhin, I.; Devlin, J.; Lee, K.; et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7: 453–466.

Levy, I.; Bogin, B.; and Berant, J. 2023. Diverse Demonstrations Improve In-context Compositional Generalization. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1401–1422. Toronto, Canada: Association for Computational Linguistics.

Li, X.; Zhao, R.; Chia, Y. K.; Ding, B.; Joty, S.; Poria, S.; and Bing, L. 2024. Chain-of-Knowledge: Grounding Large Language Models via Dynamic Knowledge Adapting over Heterogeneous Sources. In *International Conference on Learning Representations ICLR 2024*.

Ling, W.; Yogatama, D.; Dyer, C.; and Blunsom, P. 2017. Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems. In Barzilay, R.; and Kan, M.-Y., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL)*, 158–167. Vancouver, Canada: Association for Computational Linguistics.

Madaan, A.; Hermann, K.; and Yazdanbakhsh, A. 2023. What Makes Chain-of-Thought Prompting Effective? A Counterfactual Study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 1448–1535.

Miglani, V.; Yang, A.; Markosyan, A. H.; Garcia-Olano, D.; and Kokhlikyan, N. 2023. Using Captum to Explain Generative Language Models. arXiv:2312.05491.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; and Zettlemoyer, L. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work? In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11048–11064. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Patel, A.; Bhattamishra, S.; and Goyal, N. 2021. Are NLP Models really able to Solve Simple Math Word Problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2080–2094. Online: Association for Computational Linguistics.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2023. Reasoning with Language Model Prompting: A Survey. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5368–5393. Toronto, Canada: Association for Computational Linguistics.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Roy, S.; and Roth, D. 2015. Solving General Arithmetic Word Problems. In Màrquez, L.; Callison-Burch, C.; and Su, J., eds., *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1743–1752. Lisbon, Portugal: Association for Computational Linguistics.

Rubin, O.; Herzig, J.; and Berant, J. 2022. Learning To Retrieve Prompts for In-Context Learning. In Carpuat, M.; de Marneffe, M.-C.; and Meza Ruiz, I. V., eds., *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2655–2671. Seattle, United States: Association for Computational Linguistics.

Shao, Z.; Gong, Y.; Shen, Y.; Huang, M.; Duan, N.; and Chen, W. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618*.

Shum, K.; Diao, S.; and Zhang, T. 2023a. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 12113–12139. Singapore: Association for Computational Linguistics.

Shum, K.; Diao, S.; and Zhang, T. 2023b. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 12113–12139. Singapore: Association for Computational Linguistics.

Srivastava, A.; Rastogi, A.; Rao, A.; Shoeb, A. A. M.; Abid, A.; Fisch, A.; Brown, A. R.; Santoro, A.; Gupta, A.; Garriga-Alonso, A.; et al. 2023. Beyond the Imitation Game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023a. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations ICLR 2023*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; and Zhou, D. 2022. Rationale-Augmented Ensembles in Language Models. arXiv:2207.00747.

Wang, X.; Zhu, W.; Saxon, M.; Steyvers, M.; and Wang, W. Y. 2023b. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; Zoph, B.; Borgeaud, S.; Yogatama, D.; Bosma, M.; Zhou, D.; Metzler, D.; Chi, E. H.; Hashimoto, T.; Vinyals, O.; Liang, P.; Dean, J.; and Fedus, W. 2022a. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*. Survey Certification.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837.

Weng, Y.; Zhu, M.; Xia, F.; Li, B.; He, S.; Liu, S.; Sun, B.; Liu, K.; and Zhao, J. 2023. Large Language Models are Better Reasoners with Self-Verification. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2550–2575. Singapore: Association for Computational Linguistics.

Wiegreffe, S.; and Pinter, Y. 2019. Attention is not not Explanation. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 11–20. Hong Kong, China: Association for Computational Linguistics.

Winter, E. 2002. The shapley value. *Handbook of game theory with economic applications*, 3: 2025–2054.

Xie, S. M.; Raghunathan, A.; Liang, P.; and Ma, T. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. In *International Conference on Learning Representations ICLR 2022*.

Xu, F. F.; Alon, U.; Neubig, G.; and Hellendoorn, V. J. 2022. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, 1–10.

Yang, Z.; Zhang, Y.; Sui, D.; Liu, C.; Zhao, J.; and Liu, K. 2023. Representative Demonstration Selection for In-Context Learning with Two-Stage Determinantal Point Process. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 5443–5456.

Yasunaga, M.; Chen, X.; Li, Y.; Pasupat, P.; Leskovec, J.; Liang, P.; Chi, E. H.; and Zhou, D. 2024. Large Language Models as Analogical Reasoners. In *International Conference on Learning Representations ICLR 2024*.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2023. Automatic Chain of Thought Prompting in Large Language Models. In *The Eleventh International Conference on Learning Representations ICLR 2023*.

Zhao, H.; Chen, H.; Yang, F.; Liu, N.; Deng, H.; Cai, H.; Wang, S.; Yin, D.; and Du, M. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology*, 15(2): 1–38.

Zhao, R.; Li, X.; Joty, S.; Qin, C.; and Bing, L. 2023. Verify-and-Edit: A Knowledge-Enhanced Chain-of-Thought Framework. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, 5823–5840. Toronto, Canada: Association for Computational Linguistics.

Zheng, C.; Liu, Z.; Xie, E.; Li, Z.; and Li, Y. 2023. Progressive-Hint Prompting Improves Reasoning in Large Language Models. arXiv:2304.09797.

Zhong, Q.; Wang, K.; Xu, Z.; Liu, J.; Ding, L.; Du, B.; and Tao, D. 2024. Achieving 97% on GSM8K: Deeply Understanding the Problems Makes LLMs Better Solvers for Math Word Problems. arXiv:2404.14963.