Pattern-Aware Chain-of-Thought Prompting in Large Language Models

Anonymous ACL submission

Abstract

Chain-of-thought (CoT) prompting can guide language models to engage in complex multi-step reasoning. The quality of provided demonstrations significantly impacts the success of downstream inference tasks. While existing automated methods prioritize accuracy and semantics in these demonstrations, we show that the underlying reasoning patterns play a more crucial role in such tasks. In this paper, we propose PA-CoT, a prompting method that considers the diversity of demonstration patterns. By incorporating patterns such as step length and mathematical symbols within intermediate steps, PA-CoT effectively mitigates the issue of bias induced by demonstrations and enables better generalization to diverse scenarios. We conduct experiments on six reasoning benchmark tasks using two open-source LLMs. The results show that our method substantially enhances reasoning performance and exhibits robustness to errors. The code will be made available upon acceptance.

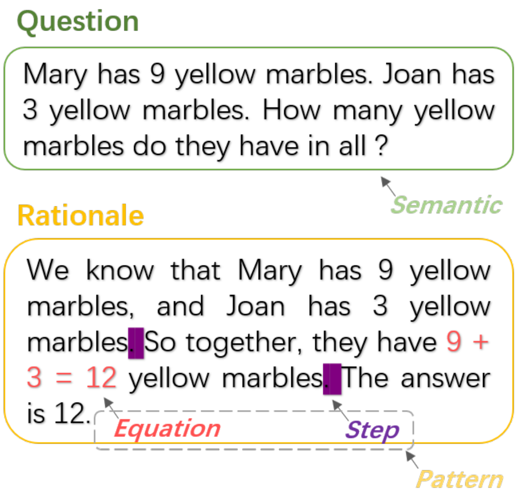

Question
Mary has 9 yellow marbles. Joan has 3 yellow marbles. How many yellow marbles do they have in all ?
Semantic
Rationale
We know that Mary has 9 yellow marbles, and Joan has 3 yellow marbles. So together, they have 9 + 3 = 12 yellow marbles. The answer is 12.
Equation
Step
Pattern

Figure 1: Example of the chain-of-thought reasoning process: This comprises a question accompanied by a rationale. The rationale serves as a depiction of how LLMs navigate the reasoning process to arrive at the answer to the given question.

1 Introduction

Large language models (LLMs) have been proven highly effective in solving complex reasoning tasks. One technique contributing to their success is the chain-of-thought (CoT) prompting (Wei et al., 2022b), which motivates the LLMs to perform multi-step reasoning instead of providing direct answers. This approach can significantly enhance the model's ability to handle challenging tasks such as arithmetic and commonsense questions.

Generally, the overall effectiveness of CoT relies on the quality of the demonstrations provided. When confronted with no examples but only the prompt "*Let's think step by step*", known as **Zero-Shot-CoT** (Kojima et al., 2022), LLMs struggle with reasoning and encounter hallucination-related issues. While manually designing demonstrations for each question can alleviate such problems (Wei et al., 2022b), it comes with a significant labour cost. To address such challenges, Zhang et al. (2023) propose **Auto-CoT**, which can automatically construct demonstrations as prompts. It initially partitions questions from a given dataset into clusters and then selects a representative question from each cluster. The selected questions are answered using Zero-Shot-CoT to obtain their **rationales** (the intermediate reasoning chain). The performance of this automated method is comparable to that of Manual-CoT.

Despite the efficacy of the automated method, how to develop a sound and complete set of demonstrations remains an area for further exploration. Several studies advocate for incorporating external knowledge to ensure the accuracy of the intermediate reasoning chain (Zhao et al., 2023; Weng et al., 2023; Li et al., 2024). Others suggest generating multiple CoT paths, complemented by a verification process to maintain self-consistency (Wang et al., 2023b; Yao et al., 2023; Liu et al., 2023).

However, most prior research focuses on the precision of demonstrations, with limited exploration of the distributional power inherent in these demonstrations. Enlightened by Min et al. (2022) and Madaan et al. (2023), LLMs perform CoT through a counterfactual approach: it does not necessitate precise example results but rather learns from the underlying **patterns** (e.g. equations, templates) exhibited by the examples.

In this paper, we introduce a novel approach called Pattern-Aware Chain-of-Thought (**PA-CoT**) and demonstrate that LLMs can achieve improved reasoning performance by embracing the diversity inherent in demonstration patterns. Following the Auto-CoT schema, we automatically generate question clusters and select representative questions from each cluster. However, instead of relying solely on question embeddings for clustering, we explore multiple methods to enrich the diversity of rationale patterns. We contend that the conventional embedding-based clustering focuses solely on question semantics, lacks reflection on the rationale, and consequently fails to encompass the full spectrum of demonstrations, as shown in Figure 1. To quantify the diversity of patterns, we introduce three metrics: **(i)** the length or steps of the rationale, where a shorter rationale implies a simpler solution, while a longer one indicates more complex reasoning requirements; **(ii)** the mathematical symbols within the rationale, where distinct equations represent different solving approaches; and **(iii)** a combination of rationale steps and symbols, providing a comprehensive perspective that considers both aspects simultaneously.

We evaluate the performance of PA-CoT across six arithmetic reasoning tasks. The experimental results consistently demonstrate that the combination strategy outperforms other methods across two LLMs. This suggests that LLMs derive substantial benefits from the diverse patterns presented in demonstrations. Further experiments are conducted to examine the impact of rationale step and symbol aspects. We empirically find that PA-CoT introduces less bias to the generated answer and exhibits error robustness, attributed to our strategy emphasizing diversity.

2 Related Work

This section reviews how chain-of-thought (CoT) prompting works and introduces various advanced approaches.

2.1 Chain-of-Thought Prompting

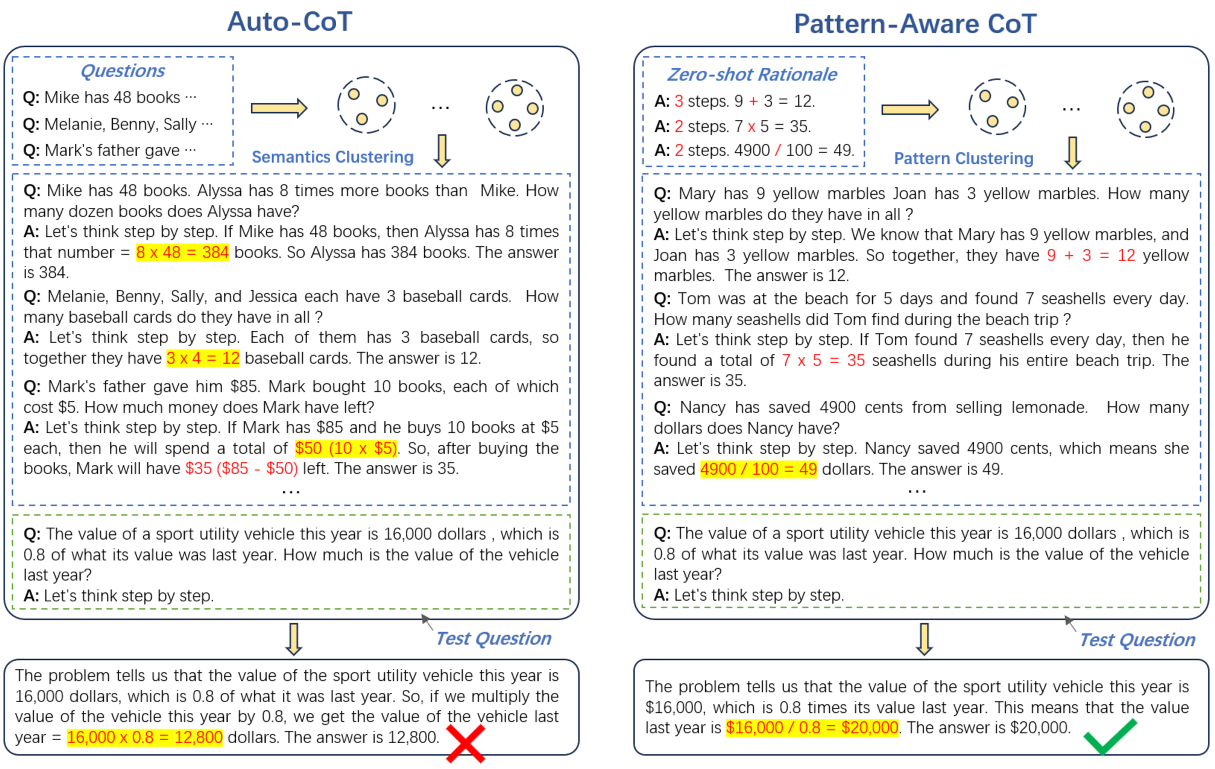
Large language models have demonstrated significant ability in comprehending context and responding to prompts (Brown et al., 2020; Ouyang et al., 2022). Recent studies highlight that LLMs can achieve improved task completion without fine-tuning, particularly on reasoning tasks, when provided with few-shot 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.

2.2 Example Selection and Refinement

Several CoT studies are directed towards automating the generation of demonstrations, such as retrieval-based (Rubin et al., 2022), zero-shot (Kojima et al., 2022), clustering-based (Zhang et al., 2023), and self-prompt (Shao et al., 2023). 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., 2023b), 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.

2.3 Role of Example Pattern

To understand the underlying mechanism of CoT, Min et al. (2022) and Madaan et al. (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



Q: Mike has 48 books ···
Q: Melanie, Benny, Sally ···
Q: Mark's father gave ···

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: Melanie, Benny, Sally, and Jessica each have 3 baseball cards. How many baseball cards do they have in all ?
A: Let's think step by step. Each of them has 3 baseball cards, so together they have 3 x 4 = 12 baseball cards. The answer is 12.
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: 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.

A: 3 steps. 9 + 3 = 12.
A: 2 steps. 7 x 5 = 35.
A: 2 steps. 4900 / 100 = 49.

Q: Mary has 9 yellow marbles Joan has 3 yellow marbles. How many yellow marbles do they have in all ?
A: Let's think step by step. We know that Mary has 9 yellow marbles, and Joan has 3 yellow marbles. So together, they have 9 + 3 = 12 yellow marbles. The answer is 12.
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, 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: Example of Auto-CoT and PA-CoT. The upper part comprises selected demonstrations and a test question, and the lower part displays the corresponding answer generated by the same LLM.

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.

3 Pattern-Aware Chain-of-Thought

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 CoT. 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 (Kojima et al., 2022) 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 et al., 2023) - to evaluate its potential benefits for the CoT process.

We argue that demonstrations function as templates, and they provide accessible reasoning formats for LLMs to emulate. The homogeneity in demonstrations poses a risk of introducing bias into the generated answers (Wang et al., 2023a). Conversely, maintaining diverse demonstrations enables a broader exploration of new reasoning inferences. Although Auto-CoT claims to cluster based on diversity, it predominantly clusters by question semantics, providing limited assistance in problem-solving. In light of this, we propose **Pattern-Aware Chain-of-Thought (PA-CoT)** that refines the example selection process to enhance the variety of reasoning chains. This approach ensures that selected examples contribute to a broader range of cases, fostering more generalizable outcomes.

In particular, we choose to experiment with arithmetic problems since the symbol patterns are relatively intuitive. Given a dataset, each question is first answered by adding the phrase "*Let's think step by step*" (zero-shot). Then we select k questions along with their rationales to serve as a general demonstration prompt for the entire dataset (Wei et al., 2022b; Zhang et al., 2023). We design a rationale-based demonstration selection method followed by three simple yet efficient variants to form our testbed:

- **Cluster based on rationale semantics.** This approach involves a straightforward shift from question embeddings to rationale embeddings. The goal is to determine if the underlying pattern can be discovered through this minor alteration. However, our experiment indicates that this method can still be distracted from irrelevant elements such as characters or scenes, hindering its ability to generate diverse demonstrations.

- **Cluster based on rationale step length.** This approach is inspired by the notion of reasoning complexity (Fu et al., 2022; Zhou et al., 2022), where a simple task typically involves a few steps, and a complex task requires longer reasoning chains. Our aim is for the demonstrations to encompass both aspects simultaneously. For instance, if all demos are complex, the test question may involve an unnecessarily lengthy reasoning process, and vice versa. To validate this hypothesis, we include two comparative studies in our experiment.

- **Cluster based on symbols in the rationale.** This approach is specifically designed for mathematical problems, where explicit equations are prevalent. In these problems, a symbol can effectively represent a solution for a particular question type. For example, an equation like $2 + 3 = 5$ can evoke the association of addition, but it provides little assistance in understanding multiplication. Our findings demonstrate that diverse symbol patterns can significantly mitigate bias in the rationale, as illustrated in Figure 2.

- **Combination of step length and symbols.** Given that the previously mentioned methods focus on distinct dimensions of rationale patterns, this approach seeks to integrate them, offering a comprehensive perspective. As semantics may introduce irrelevant distractions, it is not considered in this method. There are various ways to combine step length and symbols, and we opt for the straightforward concatenation of the two dimensions. We also test additional variants in subsequent experiments.

In summary, we adopt the aforementioned methods as our demonstration clustering strategy. We explicitly extract patterns for each question-rationale pair and encode them into vector representations using Sentence-BERT (Reimers and Gurevych, 2019). For instance, we encode "3" if the step length is 3 (split by ". " or "\n"), encode "+" if the symbol appears in the rationale (concatenate if there are multiple symbols), and encode "3 +" for our combination strategy. These representations undergo processing by the k-means clustering algorithm, similar to Auto-CoT. Within each cluster, we sort the distances and select the example closest to the centre. It is important to note that Wei et al. (2022b) and Zhang et al. (2023) both impose restrictions on the chosen example, requiring it to be simple (question less than 60 tokens and rationale less than 5 steps). In contrast, we do not impose such restrictions to preserve variety. The k selected question-rationale pairs are then assembled as the final prompt for inference.

4 Experiments

In this section, our objective is to evaluate the effectiveness of our proposed PA-CoT and assess whether the introduced variety yields benefits.

4.1 Experimental Setup

Datasets. We adopt six arithmetic problem 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), and SVAMP (Patel et al., 2021). 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., 2023a; Fu et al., 2022).

Language Models. We consider open-source large language models as our inference engine. Specifically, we choose LLaMA-2-7b-chat-hf (Touvron et al., 2023) and qwen-7b-chat (Bai et al., 2023) models, as they have been reported to be comparable to GPT-3.5[1] in terms of arithmetic ability and possess chain-of-thought reasoning capabilities. These LLMs are deployed on our local server equipped with 8x NVIDIA GeForce RTX 3090.

We use the inference function of these models and the process does not involve training or fine-tuning. We set the hyperparameter temperature as 0.4 to regulate the model's randomness (Xu et al., 2022).

It is noteworthy that, as highlighted by Wei et al. (2023), larger models are more susceptible to the

[1]https://platform.openai.com/docs/models

Table 1: Accuracy (%) on six arithmetic reasoning datasets.

Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP
LLaMA-2-7b-chat-hf	Zero-Shot-CoT	72.33	21.00	57.97	24.01	57.67	41.90
	Auto-CoT	76.00	27.36	58.48	24.01	64.96	43.80
	PA-CoT-semantic	74.83	26.76	63.29	24.80	66.92	47.19
	PA-CoT-step	76.16	24.41	**67.59**	29.13	66.14	47.59
	PA-CoT-symbol	**79.66**	25.39	65.06	25.19	**71.85**	48.50
	PA-CoT-concat	76.67	**28.05**	66.83	**29.92**	71.06	**50.10**
qwen-7b-chat	Zero-Shot-CoT	87.33	42.83	54.93	**35.03**	69.09	55.70
	Auto-CoT	90.66	47.01	62.53	30.31	80.31	60.19
	PA-CoT-semantic	**91.33**	44.80	65.06	31.88	78.74	59.00
	PA-CoT-step	90.33	46.85	**74.17**	33.07	78.14	62.00
	PA-CoT-symbol	90.50	47.16	67.59	29.52	82.08	61.50
	PA-CoT-concat	**91.33**	**48.14**	72.40	33.46	**83.85**	**62.30**

influence of examples. We observe that these 7B models can also be impacted. Thus, PA-CoT is expected to be effective in enhancing their performance.

Baselines. We primarily compare our methods with Zero-Shot-CoT (Kojima et al., 2022) and Auto-CoT (Zhang et al., 2023). To clarify the different variations of our proposed PA-CoT method, we note each pattern at the end of its name. For example, PA-CoT-semantic for clustering based on rationale semantics, and similarly for PA-CoT-step, PA-CoT-symbol, and PA-CoT-concat.

4.2 Main Results

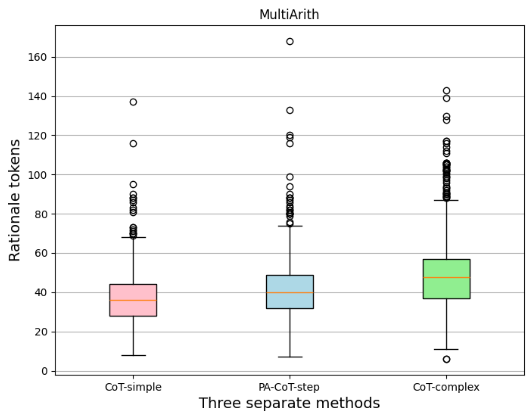
Table 1 displays the overall performance of various methods on two LLMs. Since our primary focus is on evaluating the effectiveness of PA-CoT, we are not concerned with determining which LLM outperforms the other. Based on the results, we have the following observations:

- Auto-CoT consistently outperforms Zero-Shot-CoT, indicating that the cluster-sample strategy is effective across different LLMs. With the guidance of demonstrations, LLMs exhibit an enhanced capability to generate improved results.
- Simply switching from question embeddings (Auto-CoT) to rationale embeddings (PA-CoT-semantic) does not yield significant benefits, as they generally perform at a similar level. We attribute this phenomenon to the inherent similarity between the two embeddings. As the embedding encoder considers the entire sentence as input, it unavoidably incorporates numerous irrelevant elements, such as characters and scenes. Consequently, this approach does not effectively address the fundamental problem.
- Considering naive rationale patterns (PA-CoT-step and PA-CoT-symbol) can notably enhance performance in various scenarios, with some instances even ranking first among all methods. This observation suggests that by incorporating diverse patterns into demonstrations, LLMs can effectively learn from this variability and generalize better across the entire dataset. However, given the inherent characteristics of different datasets, a single pattern may not universally adapt to every case, leading to occasional failures.
- Concatenating step length and symbol patterns (PA-CoT-concat) consistently produces the most favourable results across various scenarios compared to alternative methods. This finding implies that LLMs derive substantial benefits from incorporating multiple dimensions in the demonstration. The inclusion of both step length and symbol patterns encompasses a broader spectrum of the data distribution. Consequently, they are less prone to sampling similar examples, contributing to improved overall performance.

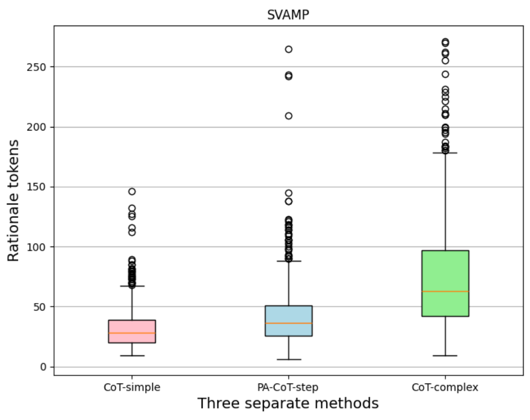
In summary, we present different approaches and evaluate their performance on arithmetic reasoning tasks. The results indicate the significance of demonstration patterns.

Table 2: Comparison between methods with various demonstration lengths.

Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP
qwen-7b-chat	PA-CoT-step	90.33	46.85	74.17	33.07	78.14	62.00
	CoT-simple	84.50	43.82	70.37	27.55	80.31	62.00
	CoT-complex	81.50	41.16	74.43	OOM	78.14	59.40



(a) MultiArith



(b) SVAMP

Figure 3: The box plot of generated rationale length across CoT-simple (pink), PA-CoT-step (blue), CoT-complex (green). The x-axis represents method names, and the y-axis represents the number of sentence tokens. The box in the middle represents where half of the numbers are. Extending from the box are whiskers that reach out to the minimum and maximum values within a specific range. Circles denote outliers, and the line splitting the box represents the median.

4.3 Impact of Step Length

To explore the influence of step length on LLMs' inference, we conduct additional experiments on this factor. In particular, we introduce two comparison methods: CoT-simple and CoT-complex. CoT-simple involves selecting examples with the fewest rationale steps, while CoT-complex involves selecting examples with the most (Fu et al., 2022). We aim to assess whether our PA-CoT-step method outperforms these two comparison methods.

Table 2 illustrates the performance of PA-CoT-step alongside two comparison methods. Overall, PA-CoT-step demonstrates advantages over the other two methods in most scenarios. We observe that CoT-complex tends to generate more errors during long intermediate steps and faces an out-of-memory (OOM) issue when the input becomes excessively long. While CoT-simple yields decent results in specific cases, it struggles with tasks requiring intricate reasoning.

We further visualize the distribution of generated answer length as in Figure 3. The box in the middle represents the interquartile range (IQR) and encapsulates the middle 50% of the data, with its lower and upper boundaries marked by the first quartile (Q1) and third quartile (Q3), respectively (Williamson et al., 1989; Kampstra, 2008). Inside the box, a line denotes the median (Q2) and indicates the dataset's central tendency. Extending from the box are whiskers that reach out to the minimum and maximum values within a specific range. Individual points beyond the whiskers signify potential outliers in the dataset.

The plot illustrates the correlation between the length of demonstrations and the number of generated tokens. With predominantly short demonstrations, CoT-simple tends to produce concise answers, resulting in a lower average value. Conversely, CoT-complex encourages longer answers, with most taking an extended route to complete the task. PA-CoT-step, on the other hand, maintains a moderate average rationale length. It covers a wider range from simple to complex reasoning. This adaptability allows it to perform well in more general situations.

Table 3: Comparison between methods with various combination strategies.

Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP
LLaMA-2-7b-chat-hf	PA-CoT-concat	76.67	28.05	66.83	29.92	71.06	50.10
	PA-CoT-sep	76.16	26.09	66.58	25.19	68.91	49.70
	PA-CoT-mean	75.83	27.67	68.86	24.01	70.86	48.19

4.4 Impact of Symbol Patterns

To investigate the role of symbol patterns in demonstrations, we also perform additional experiments on this aspect. Specifically, we categorize answers from Auto-CoT and PA-CoT-symbol based on basic arithmetic symbols: Addition, Subtraction, Multiplication, and Division. We then tally the number of correct and incorrect instances within each group. Figure 4 presents a comparison of the results on datasets AddSub and SingleEq, where the tasks are relatively straightforward.

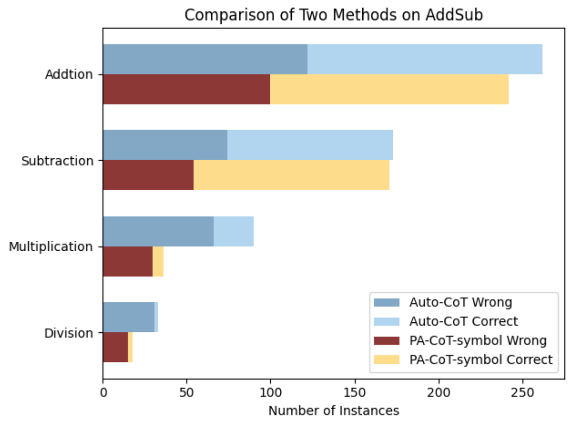
Our observations reveal that Auto-CoT produces more incorrect arithmetic equations, leading to a higher error rate within each symbol group. This indicates a higher likelihood of being misled by the demonstrations. For instance, as depicted in Figure 2, the selected demos for Auto-CoT exhibit an overemphasis on multiplication. This trend is reflected in the results of Figure 4, where Auto-CoT generates instances solved using multiplication even when it is not appropriate. In contrast, PA-CoT-symbol exhibits a better ability to select the correct solving approach, resulting in fewer errors within each group.

4.5 Combination Strategy

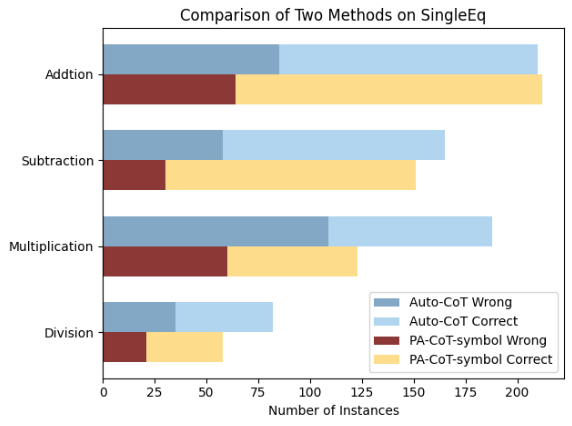
The preceding sections showcase the impact of different pattern aspects. We now turn our attention to exploring the optimal way to combine them. We initially devise PA-CoT-concat to encode the concatenation of step length and symbol strings. Considering the potential limitations of this approach, we introduce two alternative methods to explore potential improvements. The first approach involves concatenating separate vector representations encoded from step length and symbol strings, denoted as PA-CoT-sep. The second approach employs mean pooling over the separate vector representations, denoted as PA-CoT-mean. All other settings remain constant as we conduct experiments on LLaMA-2-7b-chat-hf.

Table 3 presents the comparison results of these combination strategies. Overall, the performance of PA-CoT-concat slightly exceeds that of PA-CoT-sep and PA-CoT-mean. We attribute this outcome to the different practices of semantics encoding. PA-CoT-concat takes the entire pattern string as input, where the encoded vector reflects an integration of information. In contrast, the other two approaches separate the two patterns into distinct vectors, which creates a gap between their distributions.

In conclusion, our exploration of PA-CoT and its combination strategies sheds light on the importance of considering diverse demonstration patterns in enhancing language models' reasoning ca-



(a) AddSub



(b) SingleEq

Figure 4: The distribution of the number of correct and wrong instances regarding different arithmetic symbols.

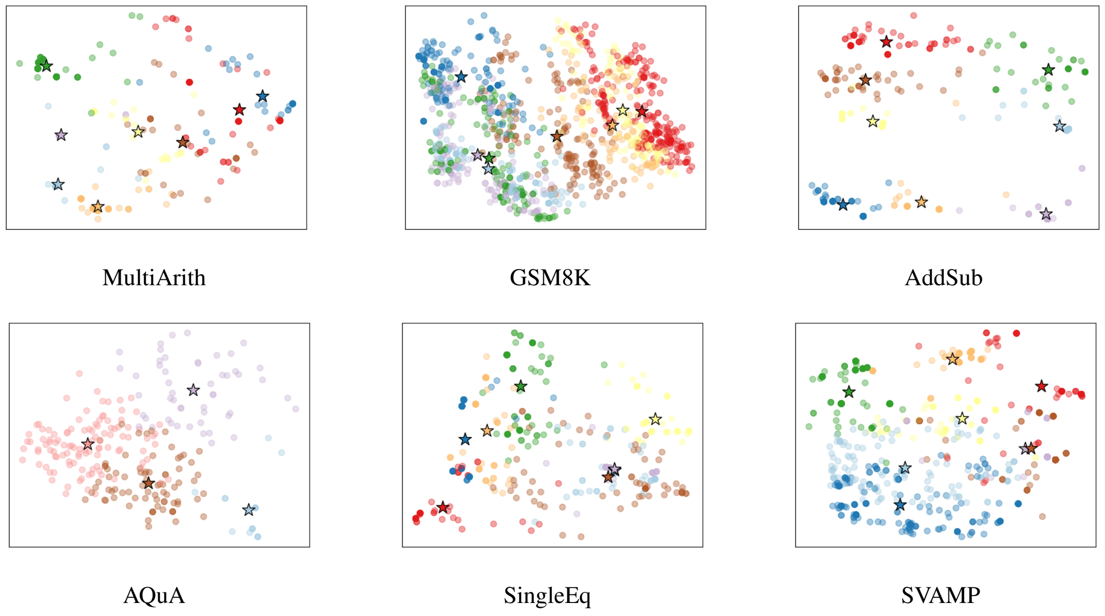


Figure 5: Visualization of clustering on six reasoning tasks. Cluster centres are noted as stars. The scatter of PA-CoT-concat clusters shows its superiority in example differentiation.

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	3	37.5%

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

pabilities. Despite slight variations in performance among the approaches, our findings underscore the significance of integrating multiple pattern aspects for improved reasoning outcomes.

4.6 Error Robustness

It is noteworthy that we do not enforce accuracy constraints on demonstrations. We proceed to count the incorrect instances within our selected demonstrations, as illustrated in Table 4.

It is intriguing to notice that the majority of our provided prompts are imperfect, with AQuA even exhibiting a 100% error rate. This phenomenon suggests that LLMs struggle to discern incorrect examples from correct ones. Instead, they learn from how the example approaches problem-solving, which we refer to as "pattern". PA-CoT encourages LLMs to follow the most probable reasoning chain towards the final answer and thus provides a significant improvement.

4.7 Visualization

Figure 5 visualizes the k clusters of PA-CoT-concat on six reasoning tasks through PCA projection. The plot depicts that there is an apparent divergence between each cluster. The scatter implies that the step length and symbols can effectively differentiate the patterns. With such diversities, LLMs can more effectively learn from demonstrations to generalize reasoning scenarios.

5 Conclusion

This paper introduces a novel pattern-aware chain-of-thought prompting method, which significantly enhances the reasoning performance of language models. Our experiments reveal that incorporating a variety of rationale step lengths prevents LLMs from taking excessively long or short steps, thereby maintaining a balanced inference chain. Similarly, diverse symbol patterns instruct LLMs to select appropriate reasoning routes and reduce bias from singular patterns. We also introduce a combination strategy that considers both aspects simultaneously with the best performance. Further investigations show the effectiveness of our proposed strategy. Apart from performance gains, our method offers additional advantages such as ease of use and error robustness.

Limitations

Due to the shutdown of OpenAI code-davinci-002 and text-davinci-002 API, we are unable to perform experiments on their models. Since most previous works choose to experiment on these models, we seek alternative LLMs as our inference engine. The two LLMs used in this paper are open-source, CoT-capable, and comparable to code-davinci-002. We hope such a practice can help future researches.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

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

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *arXiv preprint arXiv:2311.09277*.

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

Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. 2022. Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533, Doha, Qatar. Association for Computational Linguistics.

Peter Kampstra. 2008. Beanplot: A boxplot alternative for visual comparison of distributions. *Journal of statistical software*, 28:1–9.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *International Conference on Learning Representations ICLR 2024*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

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 *The Eleventh International Conference on Learning Representations ICLR 2023*.

Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. 2023. What makes chain-of-thought prompting effective? a counterfactual study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1448–1535.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *EMNLP*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752, Lisbon, Portugal. Association for Computational Linguistics.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Synthetic prompting: Generating chain-of-thought demonstrations for large language models. *arXiv preprint arXiv:2302.00618*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023a. 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*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations ICLR 2023*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

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

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

David F Williamson, Robert A Parker, and Juliette S Kendrick. 1989. The box plot: a simple visual method to interpret data. *Annals of internal medicine*, 110(11):916–921.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference. In *International Conference on Learning Representations ICLR 2022*.

Frank F Xu, Uri Alon, Graham Neubig, and Vincent Josua Hellendoorn. 2022. A systematic evaluation of large language models of code. In *Proceedings of the 6th ACM SIGPLAN International Symposium on Machine Programming*, pages 1–10.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2023. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations ICLR 2023*.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.

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. *arXiv preprint arXiv:2205.10625*.