

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

Anonymous 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 Pattern-Aware CoT, a prompting method that considers the diversity of demonstration patterns. By incorporating reasoning patterns such as step length and operation 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 nine 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 publicly available.

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 symbolic 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

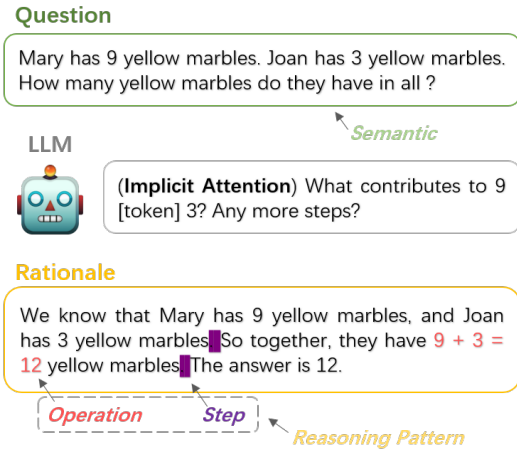


Figure 1: Example of the chain-of-thought reasoning process: This comprises a question accompanied by a rationale. The rationale serves as a depiction of how LLMs navigate the reasoning process to arrive at the answer to the given question.

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 preci-

sion 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 operations within the rationale, where distinct equations or logics represent different solving approaches; and **(iii)** a combination of rationale steps and operations, providing a comprehensive perspective that considers both aspects simultaneously.

We evaluate the performance of PA-CoT across six arithmetic and three non-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 operation 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

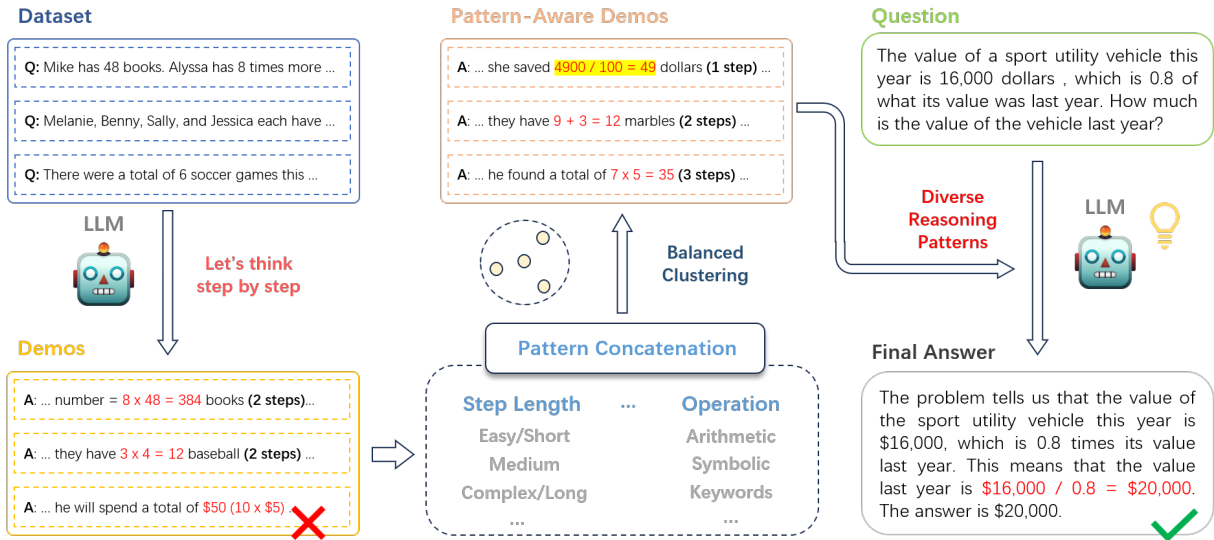


Figure 2: Illustration of Pattern-Aware Chain-of-Thought prompting. PA-CoT first extracts different patterns from original rationales. Then clustering is used to produce a group of demonstrations. PA-CoT enables LLMs to perceive diverse reasoning patterns and to select a proper solution path. It avoids LLMs being biased by monotonous reasoning mode. For a detailed case study, please refer to Appendix D.

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 and symbolic problems since the operation 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 di-

verse 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 rationale reasoning operation.** This approach is designed to extract patterns that guide the task towards reaching its objectives (Madaan et al., 2023). Empirically, we choose mathematical symbols for arithmetic tasks and keywords for symbolic ones. For more details, please see Appendix A. In these problems, an operation 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 operation patterns can significantly mitigate bias in the rationale, as illustrated in Figure 2.
- **Combination of step length and operation.** 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 the step length and the operation, 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-BERT¹ (Reimers and

¹We use all-MiniLM-L6-v2 as the embedding encoder. <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>

Gurevych, 2019). For instance, we encode “3” if the step length is 3 (split by “. ” or “\n”), encode “+” if the operation appears in the rationale (concatenate if there are multiple operations), 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 nine 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 et al., 2021), Coin-Flip (Wei et al., 2022b), BIG-bench Date Understanding, and BIG-bench Tracking Shuffled Objects (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., 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-3.5² in terms of arithmetic ability and possess chain-of-thought reasoning capabilities. These LLMs are deployed on our local server equipped with 4x NVIDIA GeForce RTX 3090. We use the inference function of these models and the process does not involve training or finetuning. 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

²<https://platform.openai.com/docs/models>

	Model	MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	Coin	Date	Tracking
LLaMA-2-7b-chat-hf	Zero-Shot-CoT	72.33	21.00	57.97	24.01	57.67	41.90	44.60	39.29	30.80
	Tab-CoT	19.16	8.56	70.63	19.69	69.35	49.80	50.40	29.81	35.19
	Auto-CoT	76.00	27.36	58.48	24.01	64.96	43.80	51.20	44.71	32.53
	(+ Self-Consistency)	82.16	32.60	59.74	22.44	69.09	52.80	57.80	40.65	30.31
	PA-CoT-semantic	74.83	26.76	63.29	24.80	66.92	47.19	48.00	43.08	31.66
	PA-CoT-step	76.16	24.41	67.59	29.13	66.14	47.59	48.00	44.44	33.33
	PA-CoT-operation	79.66	25.39	65.06	25.19	71.85	48.50	59.40	47.96	32.26
	PA-CoT-concat (+ Self-Consistency)	76.67 83.33	28.05 32.75	66.83 70.89	29.92 29.13	71.06 72.83	50.10 54.00	58.40 59.19	46.07 49.32	32.53 30.37
qwen-7b-chat	Zero-Shot-CoT	87.33	42.83	54.93	35.03	69.09	55.70	45.40	50.13	32.40
	Auto-CoT	90.66	47.01	62.53	30.31	80.31	60.19	45.40	48.78	29.73
	PA-CoT-semantic	91.33	44.80	65.06	31.88	78.74	59.00	43.20	52.38	31.00
	PA-CoT-step	90.33	46.85	74.17	33.07	78.14	62.00	38.00	49.32	30.46
	PA-CoT-operation	90.50	47.16	67.59	29.52	82.08	61.50	52.60	55.72	32.53
	PA-CoT-concat	91.33	48.14	72.40	33.46	83.85	62.30	47.40	53.13	31.60

Table 1: Accuracy (%) on nine reasoning datasets. We present the mean value obtained from five runs. For the self-consistency method, we set the number of paths as 5 (Wang et al., 2023b).

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), Tab-CoT (Ziqi and Lu, 2023), Auto-CoT (Zhang et al., 2023), and Self-Consistency (Wang et al., 2023b). 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-operation, 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-operation) 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 operation 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 operation 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 various reason-

Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	Coin	Date	Tracking
qwen-7b-chat	PA-CoT-step	90.33	46.85	74.17	33.07	78.14	62.00	38.00	49.32	30.46
	CoT-simple	84.50	43.82	70.37	27.55	80.31	62.00	47.59	52.74	32.73
	CoT-complex	81.50	41.16	74.43	OOM	78.14	59.40	38.20	39.83	31.13

Table 2: Comparison between methods with various demonstration lengths.

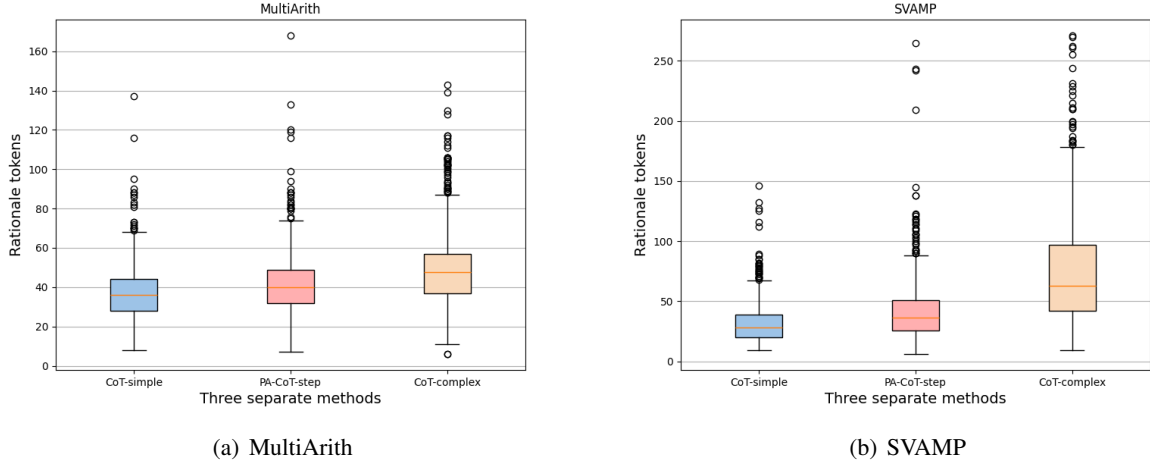


Figure 3: The box plot of generated rationale length across CoT-simple (pink), PA-CoT-step (blue), CoT-complex (green). The x-axis represents method names, and the y-axis represents the number of sentence tokens. The box in the middle represents where half of the numbers are. Extending from the box are 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.

ing tasks. The results indicate the significance of demonstration patterns.

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 gener-

ated 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.

Model		MultiArith	GSM8K	AddSub	AQuA	SingleEq	SVAMP	Coin	Date	Tracking
LLaMA-2-7b-chat-hf	PA-CoT-concat	76.67	28.05	66.83	29.92	71.06	50.10	58.40	46.07	32.53
	PA-CoT-sep	76.16	26.09	66.58	25.19	68.91	49.70	59.40	47.96	32.26
	PA-CoT-mean	75.83	27.67	68.86	24.01	70.86	48.19	54.80	41.73	31.85

Table 3: Comparison between methods with various combination strategies.

4.4 Impact of Operation

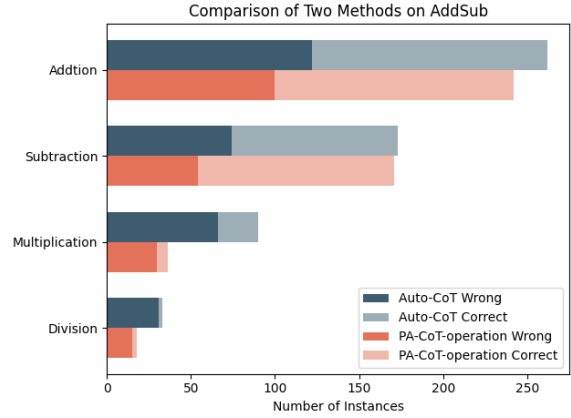
To investigate the role of operation patterns in demonstrations, we also perform additional experiments on this aspect. Specifically, we categorize answers from Auto-CoT and PA-CoT-operation based on basic arithmetic symbols: Addition, Subtraction, Multiplication, and Division. We then tally the number 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 6, 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-operation exhibits a better ability to select the correct solving approach, resulting in fewer errors within each group.

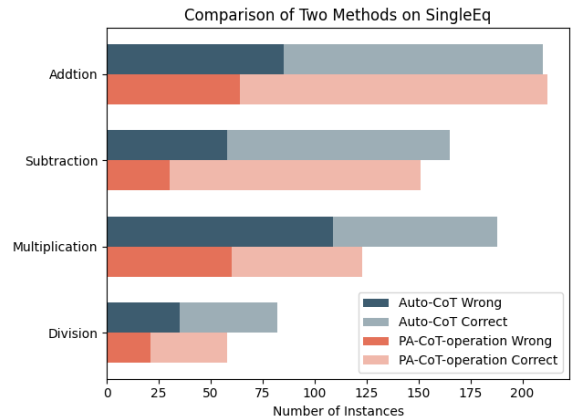
4.5 Combination Strategy

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 operation 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 operation 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



(a) AddSub



(b) SingleEq

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

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 capabilities. Despite slight variations in performance among the approaches, our findings underscore the

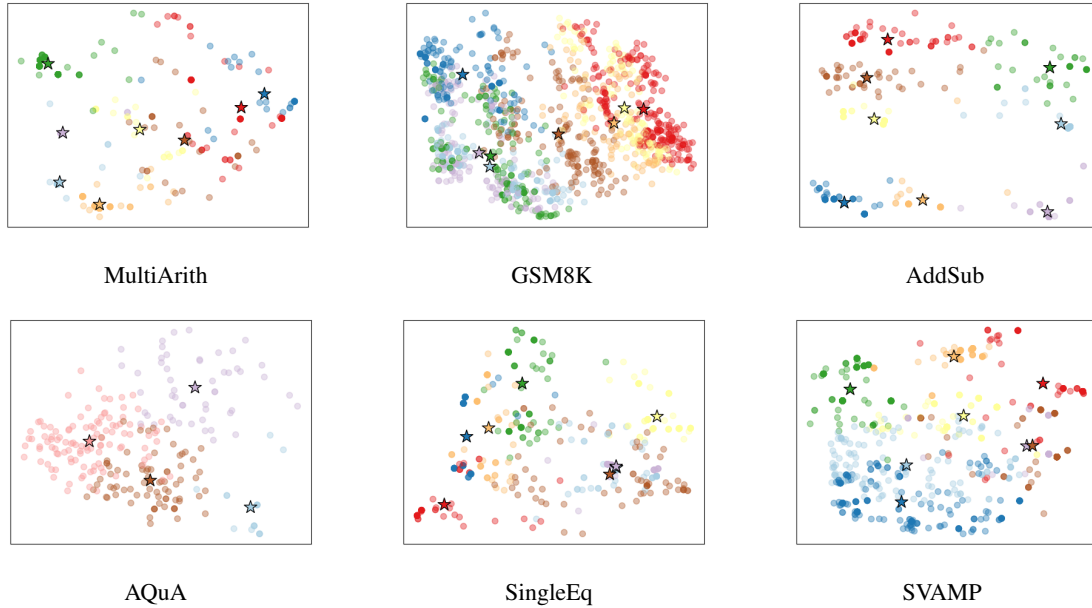


Figure 5: Visualization of clustering on six reasoning tasks. Cluster centres are noted as stars. The scatter of PA-CoT-concat clusters shows its superiority in example differentiation.

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%
Coin	8	4	50.0%
Date	8	3	37.5%
Tracking	8	4	50.0%

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

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 the operation 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 operation 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. 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.

525 Limitations

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

534 Another limitation is that our method has only
535 been tested on datasets with explicit reasoning
536 paths, such as arithmetic and symbolic tasks, where
537 patterns are intuitive and easily extractable. When
538 applied to datasets with implicit reasoning paths, it
539 may be necessary to identify the inherent reason-
540 ing operations. For more discussions, please see
541 Appendix A.

542 References

543 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang,
544 Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei
545 Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin,
546 Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu,
547 Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren,
548 Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong
549 Tu, Peng Wang, Shijie Wang, Wei Wang, Sheng-
550 guang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang,
551 Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu,
552 Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingx-
553 uan Zhang, Yichang Zhang, Zhenru Zhang, Chang
554 Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang
555 Zhu. 2023. [Qwen technical report](#). *arXiv preprint*
556 *arXiv:2309.16609*.

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

563 Yew Ken Chia, Guizhen Chen, Luu Anh Tuan,
564 Soujanya Poria, and Lidong Bing. 2023. [Con-](#)
565 [trastive chain-of-thought prompting](#). *arXiv preprint*
566 *arXiv:2311.09277*.

567 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
568 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
569 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
570 Nakano, Christopher Hesse, and John Schulman.
571 2021. [Training verifiers to solve math word prob-](#)
572 [lems](#). *arXiv preprint arXiv:2110.14168*.

573 Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark,
574 and Tushar Khot. 2022. [Complexity-based prompt-](#)
575 [ing for multi-step reasoning](#). *arXiv preprint*
576 *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 Com-
putational Linguistics. 577
578
579
580
581
582
583

Peter Kampstra. 2008. [Beanplot: A boxplot alternative](#)
[for visual comparison of distributions](#). *Journal of*
statistical software, 28:1–9. 584
585
586

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yu-
taka Matsuo, and Yusuke Iwasawa. 2022. [Large lan-](#)
[guage models are zero-shot reasoners](#). *Advances in*
Neural Information Processing Systems, 35:22199–
22213. 587
588
589
590
591

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish
Sabharwal, Oren Etzioni, and Siena Dumas Ang.
2015. [Parsing algebraic word problems into equa-](#)
[tions](#). *Transactions of the Association for Computa-*
tional Linguistics, 3:585–597. 592
593
594
595
596

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng
Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing.
2024. [Chain-of-knowledge: Grounding large lan-](#)
[guage models via dynamic knowledge adapting over](#)
[heterogeneous sources](#). In *International Conference*
on Learning Representations ICLR 2024. 597
598
599
600
601
602

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blun-
som. 2017. [Program induction by rationale genera-](#)
[tion: Learning to solve and explain algebraic word](#)
[problems](#). In *Proceedings of the 55th Annual Meet-*
ing of the Association for Computational Linguistics
(ACL), pages 158–167, Vancouver, Canada. Associa-
tion for Computational Linguistics. 603
604
605
606
607
608
609

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 *Proceedings of the 2023*
Conference on Empirical Methods in Natural Lan-
guage Processing, pages 2807–2822, Singapore. As-
sociation for Computational Linguistics. 610
611
612
613
614
615
616

Aman Madaan, Katherine Hermann, and Amir Yazdan-
bakhsh. 2023. [What makes chain-of-thought prompt-](#)
[ing effective? a counterfactual study](#). In *Findings*
of the Association for Computational Linguistics:
EMNLP 2023, pages 1448–1535. 617
618
619
620
621

Vivek Miglani, Aobo Yang, Aram H Markosyan, Diego
Garcia-Olano, and Narine Kokhlikyan. 2023. Using
captum to explain generative language models. *arXiv*
preprint arXiv:2312.05491. 622
623
624
625

Sewon Min, Xinxu Lyu, Ari Holtzman, Mikel Artetxe,
Mike Lewis, Hannaneh Hajishirzi, and Luke Zettle-
moyer. 2022. [Rethinking the role of demonstra-](#)
[tions: What makes in-context learning work?](#) In *Proceed-*
ings of the 2022 Conference on Empirical Methods in
Natural Language Processing, pages 11048–11064,
Abu Dhabi, United Arab Emirates. Association for
Computational Linguistics. 626
627
628
629
630
631
632
633

634	Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida,	Ruan Silva, Eric Michael Smith, Ranjan Subrama-	692
635	Carroll Wainwright, Pamela Mishkin, Chong Zhang,	nian, Xiaoqing Ellen Tan, Binh Tang, Ross Tay-	693
636	Sandhini Agarwal, Katarina Slama, Alex Ray, et al.	lor, Adina Williams, Jian Xiang Kuan, Puxin Xu,	694
637	2022. Training language models to follow instruc-	Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan,	695
638	tions with human feedback. <i>Advances in Neural</i>	Melanie Kambadur, Sharan Narang, Aurelien Ro-	696
639	<i>Information Processing Systems</i> , 35:27730–27744.	driguez, Robert Stojnic, Sergey Edunov, and Thomas	697
		Scialom. 2023. Llama 2: Open foundation and fine-	698
640	Arkil Patel, Satwik Bhattamishra, and Navin Goyal.	tuned chat models.	699
641	2021. Are NLP models really able to solve simple		
642	math word problems? In <i>Proceedings of the 2021</i>	Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark	700
643	<i>Conference of the North American Chapter of the</i>	Steyvers, and William Yang Wang. 2023a. Large	701
644	<i>Association for Computational Linguistics: Human</i>	language models are latent variable models: Explain-	702
645	<i>Language Technologies</i> , pages 2080–2094, Online.	ing and finding good demonstrations for in-context	703
646	Association for Computational Linguistics.	learning. In <i>Thirty-seventh Conference on Neural</i>	704
		<i>Information Processing Systems.</i>	705
647	Nils Reimers and Iryna Gurevych. 2019. Sentence-bert:	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc	706
648	Sentence embeddings using siamese bert-networks.	Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery,	707
649	In <i>Proceedings of the 2019 Conference on Empirical</i>	and Denny Zhou. 2023b. Self-consistency improves	708
650	<i>Methods in Natural Language Processing (EMNLP).</i>	chain of thought reasoning in language models. In	709
651	Association for Computational Linguistics.	<i>The Eleventh International Conference on Learning</i>	710
		<i>Representations ICLR 2023.</i>	711
652	Subhro Roy and Dan Roth. 2015. Solving general arith-	Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel,	712
653	metic word problems. In <i>Proceedings of the 2015</i>	Barret Zoph, Sebastian Borgeaud, Dani Yogatama,	713
654	<i>Conference on Empirical Methods in Natural Lan-</i>	Maarten Bosma, Denny Zhou, Donald Metzler, Ed H.	714
655	<i>guage Processing (EMNLP)</i> , pages 1743–1752, Lis-	Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy	715
656	bon, Portugal. Association for Computational Lin-	Liang, Jeff Dean, and William Fedus. 2022a. Emer-	716
657	guistics.	gent abilities of large language models. <i>Transactions</i>	717
		<i>on Machine Learning Research.</i> Survey Certifica-	718
658	Ohad Rubin, Jonathan Herzig, and Jonathan Berant.	tion.	719
659	2022. Learning to retrieve prompts for in-context	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten	720
660	learning. In <i>Proceedings of the 2022 Conference of</i>	Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou,	721
661	<i>the North American Chapter of the Association for</i>	et al. 2022b. Chain-of-thought prompting elicits rea-	722
662	<i>Computational Linguistics: Human Language Tech-</i>	soning in large language models. <i>Advances in Neural</i>	723
663	<i>nologies</i> , pages 2655–2671, Seattle, United States.	<i>Information Processing Systems</i> , 35:24824–24837.	724
664	Association for Computational Linguistics.		
665	Zhihong Shao, Yeyun Gong, Yelong Shen, Min-	Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert	725
666	lie Huang, Nan Duan, and Weizhu Chen. 2023.	Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu,	726
667	Synthetic prompting: Generating chain-of-thought	Da Huang, Denny Zhou, et al. 2023. Larger language	727
668	demonstrations for large language models. <i>arXiv</i>	models do in-context learning differently. <i>arXiv</i>	728
669	<i>preprint arXiv:2302.00618.</i>	<i>preprint arXiv:2303.03846.</i>	729
670	Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao,	Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He,	730
671	Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch,	Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao.	731
672	Adam R Brown, Adam Santoro, Aditya Gupta, Adria	2023. Large language models are better reasoners	732
673	Garriga-Alonso, et al. 2023. Beyond the imitation	with self-verification. In <i>Findings of the Associa-</i>	733
674	game: Quantifying and extrapolating the capabili-	<i>tion for Computational Linguistics: EMNLP 2023,</i>	734
675	ties of language models. <i>Transactions on Machine</i>	pages 2550–2575, Singapore. Association for Com-	735
676	<i>Learning Research.</i>	putational Linguistics.	736
677	Hugo Touvron, Louis Martin, Kevin Stone, Peter Al-	David F Williamson, Robert A Parker, and Juliette S	737
678	bert, Amjad Almahairi, Yasmine Babaei, Nikolay	Kendrick. 1989. The box plot: a simple visual	738
679	Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti	method to interpret data. <i>Annals of internal medicine,</i>	739
680	Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton	110(11):916–921.	740
681	Ferre, Moya Chen, Guillem Cucurull, David Esiobu,	Sang Michael Xie, Aditi Raghunathan, Percy Liang,	741
682	Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller,	and Tengyu Ma. 2022. An explanation of in-context	742
683	Cynthia Gao, Vedanuj Goswami, Naman Goyal, An-	learning as implicit bayesian inference. In <i>Interna-</i>	743
684	thony Hartshorn, Saghar Hosseini, Rui Hou, Hakan	<i>tional Conference on Learning Representations ICLR</i>	744
685	Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa,	2022.	745
686	Isabel Kloumann, Artem Korenev, Punit Singh Koura,	Frank F Xu, Uri Alon, Graham Neubig, and Vincent Jo-	746
687	Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Di-	sua Hellendoorn. 2022. A systematic evaluation of	747
688	ana Liskovich, Yinghai Lu, Yuning Mao, Xavier Mar-	large language models of code. In <i>Proceedings of</i>	748
689	tinet, Todor Mihaylov, Pushkar Mishra, Igor Moly-		
690	bog, Yixin Nie, Andrew Poulton, Jeremy Reizen-		
691	stein, Rashi Rungta, Kalyan Saladi, Alan Schelten,		

749 *the 6th ACM SIGPLAN International Symposium on*
750 *Machine Programming*, pages 1–10.

751 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran,
752 Thomas L. Griffiths, Yuan Cao, and Karthik
753 Narasimhan. 2023. [Tree of thoughts: Deliberate](#)
754 [problem solving with large language models](#).

755 Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex
756 Smola. 2023. [Automatic chain of thought prompting](#)
757 [in large language models](#). In *The Eleventh International*
758 *Conference on Learning Representations ICLR*
759 *2023*.

760 Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei
761 Qin, and Lidong Bing. 2023. [Verify-and-edit: A](#)
762 [knowledge-enhanced chain-of-thought framework](#).
763 In *Proceedings of the 61st Annual Meeting of the As-*
764 *sociation for Computational Linguistics (ACL)*, pages
765 5823–5840, Toronto, Canada. Association for Com-
766 putational Linguistics.

767 Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo
768 Li, and Yu Li. 2023. [Progressive-hint prompting](#)
769 [improves reasoning in large language models](#). *arXiv*
770 *preprint arXiv:2304.09797*.

771 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei,
772 Nathan Scales, Xuezhi Wang, Dale Schuurmans,
773 Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022.
774 [Least-to-most prompting enables complex reason-](#)
775 [ing in large language models](#). *arXiv preprint*
776 *arXiv:2205.10625*.

777 Jin Ziqi and Wei Lu. 2023. [Tab-CoT: Zero-shot tabular](#)
778 [chain of thought](#). In *Findings of the Association for*
779 *Computational Linguistics: ACL 2023*, pages 10259–
780 10277, Toronto, Canada. Association for Computa-
781 tional Linguistics.

782 A Operation Patterns

783 The operation pattern generally guides the task to-
784 wards reaching its objectives and hence can man-
785 ifest differently depending on the task (Madaan
786 et al., 2023). Table 5 shows the operation pattern
787 identified in our experiments.

788 For arithmetic tasks, we utilize a standardized
789 set of mathematical symbols. For symbolic tasks,
790 we identify empirical keywords as patterns, which
791 serve a similar function to symbols. For instance,
792 in scenarios like the Coin Flip dataset, consecu-
793 tive patterns such as ‘head-head-head’ are different
794 from ‘head-tail-tail’. PA-CoT incentivizes LLMs to
795 learn from these reasoning pathways. Additionally,
796 there are potential automated methods for pattern
797 selection, such as chi-square testing and keyword
798 extraction.

Dataset	Operation Pattern
MultiArith GSM8K AddSub AQuA SingleEq SVAMP	+ , - , * , x , / , % , > , <
Coin Date Tracking	‘heads up’, ‘tails up’ ‘day’, ‘week’, ‘month’, ‘year’, ‘yesterday’, ‘tomorrow’ ‘trade’, ‘switch’, ‘exchange’, ‘swap’

Table 5: The operation pattern identified for each dataset.

B Heuristic Selection

799 We aim for a fair comparison with Auto-CoT, 800
801 which utilizes embeddings for clustering. Our ex- 802
803 periments are designed to showcase that while clus- 804
805 tering question embeddings may not always iden- 806
807 tify the best examples, clustering pattern embed- 808
809 dings can achieve this to some extent. Acknowledg- 810
811 ing that embeddings might not be the optimal 812
813 clustering approach, we introduce another variant 814
815 of PA-CoT using heuristics. Specifically, we orga- 816
817 nize demonstrations into groups based on the num- 818
819 ber of reasoning steps (e.g., 1, 2, ..., k), selecting 820
821 one random instance per group instead of creating 822
823 embeddings and clusters. This approach allows us 824
825 to evaluate the effectiveness of embeddings. 826

827 Table 6 shows the comparison between PA-CoT- 828
829 step and PA-CoT-heuristic. We observe a marginal 830
831 improvement with PA-CoT-heuristic. This suggests 832
833 that using Sentence-BERT to encode information 834
835 such as step length does not significantly hinder per- 836
837 formance. However, there is potential to enhance 838
839 performance by adjusting the method of demon- 840
841 stration selection. 842

C Larger Models

823 To explore the effectiveness of our method on larger 824
825 models, we additionally conduct tests using GPT- 826
827 3.5-turbo. Table 7 presents the results of various 828
829 methods on this model. We observe that with 830
831 the use of a larger model, PA-CoT consistently 832
833 achieves better accuracy across different tasks, in- 834
835 dicated that our method is also applicable to large- 836
837 scale models. 838

D Case Study

839 To gain a deeper understanding of the working 840
841 mechanism of CoT reasoning, we conduct a case 842
843 study. We select the Captum (Migliani et al., 844
845 2023) tool for visualization. Specifically, we use 846

Auto-CoT

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 = $8 \times 48 = 384$ 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 $3 \times 4 = 12$ baseball cards. The answer is 12.

Q: There were a total of 6 soccer games this year. Jessica missed 4 of the games. How many soccer games did Jessica go to in all?
A: Let's think step by step. We know that there were 6 soccer games in total, and Jessica missed 4 of them. That means she went to $6 - 4 = 2$ games. So, Jessica went to 2 soccer games in all. The answer is 2.

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 $\$50 (10 \times \$5)$. 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 = $16,000 \times 0.8 = 12,800$ dollars. The answer is 12,800. ❌

Pattern-Aware CoT

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.

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 $7 \times 5 = 35$ 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: Tim's cat had kittens. He gave 3 to Jessica and 6 to Sara. He now has 9 kittens left. How many kittens did he have to start with?
A: Let's think step by step. We know that Tim gave 3 kittens to Jessica, so the number of kittens he had after giving them away is $9 - 3 = 6$. Then we know that he gave 6 kittens to Sara, so the number of kittens he had after giving them away is $6 - 6 = 0$. Therefore, Tim had 9 kittens to start with. The answer is 9.

...

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 6: Case study for Auto-CoT and PA-CoT on the same question. The upper part comprises selected demonstrations and a test question, and the lower part displays the corresponding answer generated by the same LLM.

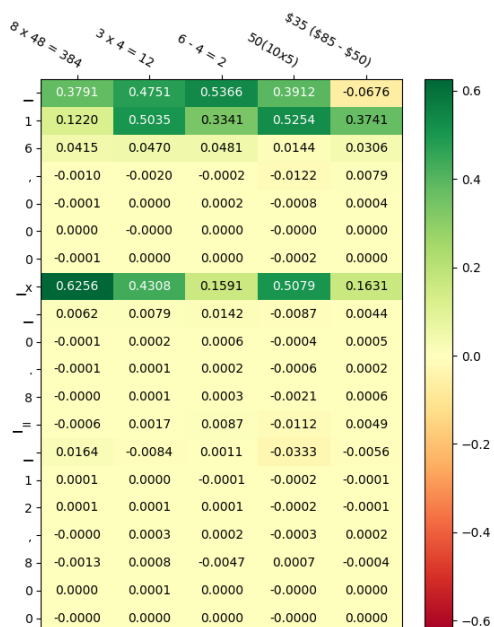


Figure 7: Visualization of token attribution for the case of Auto-CoT.

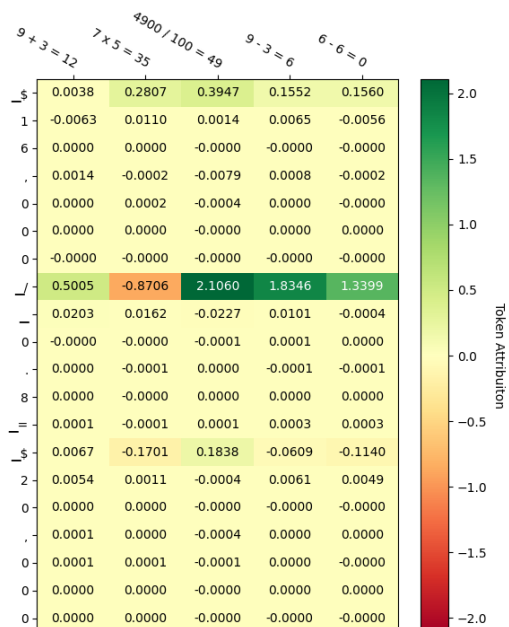


Figure 8: Visualization of token attribution for the case of PA-CoT.

Model		MultiArith	GSM8K	AddSub
LLaMA-2-7b-chat-hf	PA-CoT-step	76.16	24.41	67.59
	PA-CoT-heuristic	77.00	24.94	67.84
qwen-7b-chat	PA-CoT-step	90.33	46.85	74.17
	PA-CoT-heuristic	91.66	45.18	75.20

Table 6: Comparison between methods with and without embedding clustering.

Model		AddSub	AQuA	SingleEq
gpt-3.5-turbo-0125	Zero-Shot-CoT	83.29	59.44	90.55
	Auto-CoT	81.26	58.66	91.53
	PA-CoT-concat	83.54	62.38	93.11

Table 7: Result of GPT-3.5-turbo model.

836 perturbation-based methods from Captum to in-
837 vestigate the attribution scores of output tokens
838 to input tokens, which can intuitively help us un-
839 derstand the output logic of the black-box model.
840 Figure 6 shows a reasoning example and the re-
841 spective answers from auto-cot and pa-cot. Figures
842 7 and 8 present the tables of attribution scores of
843 the output results to the input prompts. A positive
844 score indicates that the example is beneficial for
845 token generation, while a negative score indicates
846 that the example does not help with token impact.
847 We observe that auto-cot, due to the introduction of
848 excessive bias, causes the model to focus more on
849 incorrect reasoning logic during output. In contrast,
850 PA-CoT uses more diverse examples, allowing the
851 model to learn various logical patterns better when
852 answering, thereby choosing the correct response
853 method.