Strategic Chain-of-Thought: Guiding Accurate Reasoning in LLMs through Strategy Elicitation

Anonymous ACL submission

Abstract

The Chain-of-Thought (CoT) paradigm has emerged as a critical approach for enhancing the reasoning capabilities of large language models (LLMs). However, despite their widespread adoption and success, CoT methods often exhibit instability due to their inability to consistently ensure the quality of generated reasoning paths, leading to sub-optimal reasoning performance. To address this challenge, we propose the Strategic Chain-of-Thought (SCoT), a novel methodology designed to refine LLM performance by integrating strategic knowledge prior to generating intermediate reasoning steps. SCoT employs a two-stage approach within a single prompt: first eliciting an effective problem-solving strategy, which is then used to guide the generation of high-quality CoT paths and final answers. Our experiments across eight challenging reasoning datasets demonstrate significant improvements, including a 21.05% increase on the GSM8K dataset and 24.13% on the Tracking_Objects dataset, respectively, using the Llama3-8b model. Additionally, we extend the SCoT framework to develop a few-shot method with automatically matched demonstrations, yielding even stronger results. These findings underscore the efficacy of SCoT, highlighting its potential to substantially enhance LLM performance in complex reasoning tasks.

1 Introduction

The rapid development of large language models (LLMs) has highlighted their remarkable effectiveness in reasoning tasks (Huang and Chang, 2022; Chang et al., 2024), particularly when integrated with various prompting techniques (Sivarajkumar et al., 2023). Among the techniques, Chain-of-Thought (CoT) has become a fundamental component of contemporary LLMs and is now widely adopted in the field of natural language processing.

Despite the demonstrated effectiveness of the CoT approach in various applications, it faces significant challenges in complex reasoning tasks. These challenges primarily arise from the variability in the quality of the reasoning paths generated by the CoT method (Wang et al., 2022), which are not consistently optimal. Consequently, even when LLMs produce a CoT path that aligns with a valid reasoning process, there remains a risk that the final outcome may be erroneous.

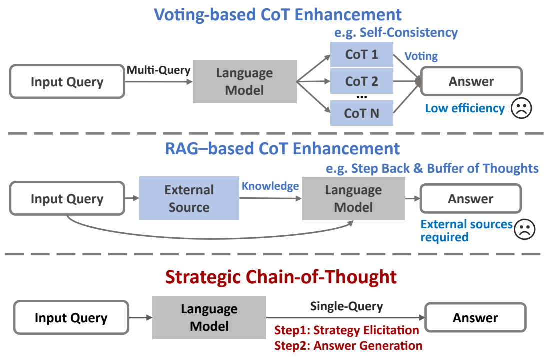


Figure 1: Comparison of some popular methods with SCoT: As a single-query method, SCoT is efficient and does not rely on external knowledge sources, distinguishing it from other approaches.

This phenomenon parallels findings in cognitive science, where different problem-solving strategies, though correct, vary in error likelihood. Sweller's Cognitive Load Theory (Sweller, 1988) suggests that different strategies impose varying cognitive loads, influencing error probability.

This variability in CoT generation strategies can reduce the reliability of CoT approaches, especially in critical applications requiring precise reasoning. Thus, further refinement is needed to enhance CoT performance in complex reasoning tasks, drawing on insights from both AI and cognitive science.

Several methods address this challenge by improving CoT path quality. Voting-based approaches improve reasoning accuracy by generating diverse paths and selecting the most reliable answer (Wang et al., 2022; Zhang et al., 2023). Retrieval-

Augmented Generation (RAG) uses multi-step prompting to access external knowledge (Lewis et al., 2021; Yang et al., 2024b; Zheng et al., 2023). Additionally, Suzgun and Kalai(2024) integrate prompt enhancement algorithms that dynamically select the optimal one during operation. Prompt-based methods guide models through predefined reasoning patterns by incorporating cue words, such as generating a plan before providing a solution(Wang et al., 2023).

These approaches help reduce variability in path quality but often come with high costs. For instance, Self-Consistency (Wang et al., 2022) may require up to 40 queries, while methods like BoT (Yang et al., 2024b) and Step-Back (Zheng et al., 2023) involve multi-stage queries. Step-Back abstracts the problem, providing more knowledge but not directly identifying key steps for solving it. Additionally, RAG-based approaches rely on high-quality external expert resources. Prompt-based methods like Plan-and-Solve (Wang et al., 2023), while optimizing CoT paths, still share similar limitations with traditional CoT methods and may result in suboptimal reasoning paths.

To tackle this challenge, we propose a novel approach called Strategic Chain-of-Thought (SCoT). SCoT is designed to improve the quality of CoT path generation for reasoning tasks by incorporating strategic knowledge. The method involves a two-step process within a single prompt. First, it explores and identifies various problem-solving strategies, eliciting the most effective one as the guiding strategic knowledge. Subsequently, this strategic knowledge directs the model in generating high-quality CoT paths and producing accurate final answers, ensuring a more effective reasoning process. We further extend the SCoT framework by adapting it to a few-shot method. In this approach, strategic knowledge is used to automatically select the most relevant demonstrations. These examples can be employed within both the few-shot and SCoT frameworks to further enhance reasoning capability. SCoT enhances the model's reasoning capabilities without the need for multi-query approaches or additional knowledge sources. By eliminating the requirement for multiple queries and external knowledge integration, SCoT reduces computational overhead and operational costs, making it a more practical and resource-efficient solution.

We conducted experiments across eight reasoning datasets spanning five distinct domains: mathematical reasoning, commonsense reasoning, physical reasoning, spatial reasoning, and multi-hop reasoning. The results revealed substantial improvements across various models, including a 21.05% increase in accuracy on the GSM8K dataset and a 24.13% increase on the Tracking_Objects dataset with the Llama3-8b model. These results validate the effectiveness of the SCoT approach.

The contributions of this work are summarized as follows:

- We introduce a two-stage methodology that integrates strategic knowledge, guiding the LLM to generate high-quality CoT paths by first developing a problem-solving strategy and then producing the final answer.
- We propose a method that leverages strategic knowledge to select and match relevant demonstrations, enabling the precise pairing of high-quality CoT examples.
- Our experimental results validate the effectiveness of SCoT, demonstrating promising outcomes in reasoning tasks across multiple domains.

2 Related Work

2.1 Strategic Diversity in Problem Solving

In the realm of problem-solving, there is rarely a one-size-fits-all approach. The complexity of each problem often necessitate a variety of strategies to reach an effective solution. In the fields of education and cognitive science, the phenomenon of using multiple approaches to solve problems is quite common (Sweller, 1988; Rusczyk, 2003). Similarly, researchers have found that LLMs might generate diverse solution paths for one question, where the problem-solving strategies and answers of these methods might vary significantly (Wang and Zhou, 2024; Wang et al., 2022).

2.2 Enhancement of CoT Path

Current methods for improving model-generated content are diverse and sophisticated.

Some approaches improve reasoning by adding specific phrases to prompt templates. For example, some methods assigned roles at the start of the prompt to elicit more professional responses (Kong et al., 2024), while others used techniques like "Take a breath" (Yang et al., 2023) or prompting the model to first create a plan or a principle before solving the problem (Zheng et al., 2023; Wang et al., 2023) to generate higher-quality CoT paths.

Voting-based mechanisms have gained prominence in recent research efforts. Wang et al. (2022) proposed the Self-Consistency, which enhances reasoning accuracy by generating over 20 CoT paths and then selecting the most consistent answer through voting. Similarly, Zheng et al. (2023) introduced Step-Back, which enhances RAG by abstracting the question for better logical structure. Similarly, Yang et al. (2024b) developed another RAG-based method, Buffer of Thoughts, integrating external knowledge into task-specific prompt templates to generate more accurate answers.

Additionally, some methods incorporate external tools for problem-solving. PAL (Gao et al., 2023) utilized LLMs to parse problems and generated programs as intermediate reasoning steps, delegating the execution of solutions to a runtime environment such as a Python interpreter. Tree of Thoughts (ToT) (Yao et al., 2024) introduced a tree-based reasoning structure to enhance decision-making processes and to improve reasoning capabilities. Suzgun and Kalai (2024) introduced meta-prompting, a technique that integrates existing prompt-based frameworks to enable dynamic selection of the most effective reasoning strategy.

These methods are complex, with some being task-specific and others requiring multi-turn prompting. However, they have proven effective in enhancing LLM reasoning, advancing CoT generation in machine learning.

3 Method

In this section, we introduce the strategic knowledge, the Strategic Chain-of-Thought (SCoT) method, and its few-shot extension.

3.1 Strategic Knowledge

LLMs tend to generate different CoT paths for the same problem, but their quality can vary significantly (Wang and Zhou, 2024; Wang et al., 2022). As shown in Figure 2(a), even methods like Plan-and-Solve, known for higher accuracy, can produce errors when solving problems like the math question "*compute the sum of all integers s such that* $-26 < s < 24$". An alternative approach, using the arithmetic series sum formula, provides more stable and accurate results. While both methods are valid, the formula-based approach leads to higher-quality, more stable outputs and is considered strategic knowledge.

Strategic knowledge (Strategy) refers to a well-defined method or principle that guides reasoning towards a correct and stable solution. It involves using structured processes that logically lead to the desired outcome, thereby enhancing the stability of CoT generation and improving the overall quality of the results.

Specifically, strategic knowledge should adhere to the following principles:

1. Correct and Comprehensive Problem-Solving Approach: The strategy provides a systematic approach that allows the model to generate accurate answers when it follows the reasoning steps.

2. Straightforward Problem-Solving Steps: The steps of the method should not be overly complex, while each step should be sufficiently detailed to ensure accuracy and prevent overly brief outputs that could lead to ambiguity.

3.2 Strategic Chain-of-Thought

Building on the concept of strategic knowledge, we propose a prompt-based method to enhance the reasoning quality of LLMs, called Strategic Chain-of-Thought (SCoT).

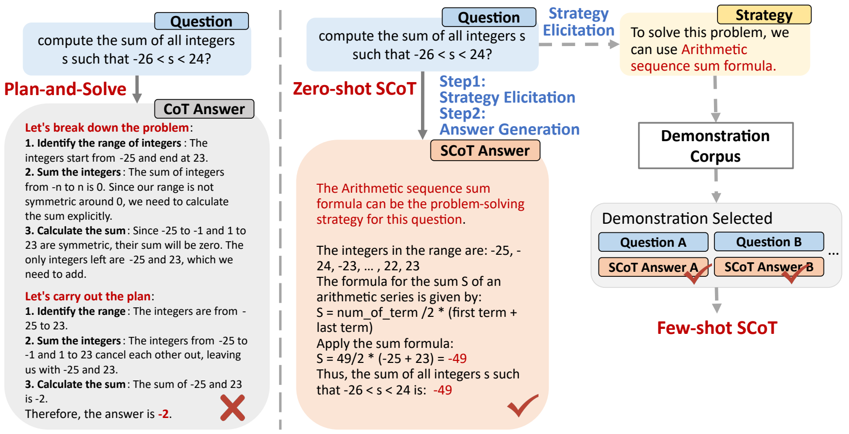
The SCoT method enables the model to first elicit strategic knowledge before generating an answer, rather than producing an answer directly. Specifically, in a single-query setting, SCoT involves two key steps:

1. **Elicitation of Strategic Knowledge**: The model identifies and determines one of the most effective methods for solving the problem, which then serves as the strategic knowledge for the task.

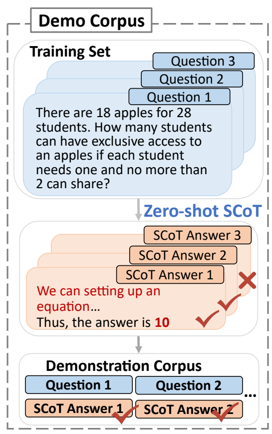
2. **Application of Strategic Knowledge**: The model subsequently applies the identified strategy to solve the problem and derive the final answer.

Figure 3(a) illustrates a prompt template utilizing the SCoT approach. Our prompt consists of five components: Role, Workflow, Rule, Initialization, and Task Input. The workflow, as shown in Figure 4, comprising three steps integrated into a single prompt. The first two steps are designed to identify and elicit strategic knowledge for solving the problem, while the third step focuses on applying the strategy to generate the answer.

We demonstrate that the details of strategic knowledge identification vary across different domains. In mathematics, strategic knowledge favors generating elegant and efficient solutions, such as using the arithmetic series formula to sum sequences. In physics, it involves selecting the most relevant and straightforward formulas or processes, such as applying $F = ma$ to calculate force. For

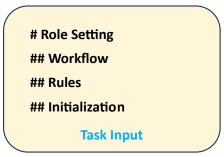


(a) Framework of Zero-shot and Few-shot Strategic Chain-of-Thought. The solid line in the middle represents Zero-shot SCoT, while the dashed line on the right represents Few-shot SCoT. Details of the Prompt in Methods are omitted.

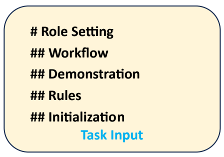


(b) Construction of the Demonstration Corpus

Figure 2: Illustration of Zero-shot and Few-shot Strategic SCoT. Few-shot SCoT builds upon Zero-shot SCoT by incorporating selected demonstrations. Details of the Few-shot SCoT approach are omitted due to space limitations.



(a) SCoT



(b) Few-shot SCoT

Figure 3: Prompt templates for zero-shot SCoT and few-shot SCoT

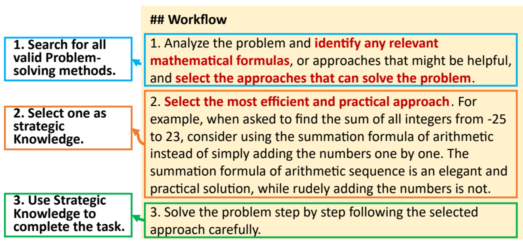


Figure 4: Example of a workflow in a Math task prompt

multi-hop reasoning, strategic knowledge focuses on determining the appropriate granularity for problem decomposition and recalling pertinent information. Similarly, in other domains, the model first develops an overarching method or workflow before systematically applying it to solve problems.

3.3 Few-shot Strategic Chain-of-Thought

We refine the SCoT method into a few-shot version by leveraging the strategy to select demonstrations. Our approach is structured into two stages: constructing a strategy-based demonstration corpus (Figure 2(b)) and performing model inference in a two-query process (Figure 2(a)). This is motivated by the fact that some problems, despite being from different domains, share similar solution strategies. As a result, direct similarity matching based on the problems alone may not yield the most relevant demonstrations.

Stage 1: Strategic Knowledge-Based Demonstration Corpus Construction.

1. SCoT Answer Generation: We apply zero-shot SCoT to the instances in the training set to generate SCoT answers for each question.

2. Demonstration Corpus Construction: The generated answers are compared to the ground truth, retaining only correct question-SCoT pairs. This step assumes that the strategic knowledge used in these problems is both correct and relevant. The validated pairs are then compiled into a demonstration corpus based on the strategic knowledge.

Stage 2: Model Inference.

1. Strategic Knowledge Generation: The LLM generates strategy for each test instance, focusing on understanding the problem rather than providing the final answer.

2. Demonstration Matching: The strategy is used to search the demonstration corpus from Stage 1, matching relevant demonstrations to the SCoT answers. This is done by computing embeddings with the m3e-base model and selecting the most similar examples from the corpus.

3. Few-shot Inference: The selected demonstrations are integrated as few-shot examples into the input prompt (Figure 3(b)). This integration guides the model to generate the final prediction based on

the provided examples.

4 Experimental Setup

In this section, we introduce the detailed experimental setup for validation of SCoT, including the datasets used for testing, the models covered, and the baselines employed.

4.1 Datasets and Tasks

To validate the effectiveness of the SCoT method, we collect a range of reasoning-related datasets, covering domains including mathematics and physical reasoning, commonsense and multi-hop reasoning, and spatial reasoning:

1. Mathematics and Physical Reasoning: We assess the models using datasets such as MathQA (Amini et al., 2019), AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021), and MMLU-high-school-math (Hendrycks et al., 2021) for mathematical reasoning tasks. These datasets feature a range of mathematical problems with varying levels of difficulty, demanding strong mathematical reasoning abilities. Additionally, we evaluated the models on ARC_Challenge (Clark et al., 2018) for physical reasoning, *i.e.*, a popular dataset that presents significant challenges in this domain.

2. Commonsense and Multi-hop Reasoning: We evaluate the models on CommonsenseQA (CSQA) (Talmor et al., 2019) for commonsense reasoning tasks and StrategyQA (SQA) (Geva et al., 2021) for multi-hop reasoning tasks. These datasets are well-regarded in their respective domains and offer a substantial level of difficulty.

3. Spatial Reasoning: We also evaluate the models using the Tracking_Object (Object) (BIG-bench authors, 2023) dataset, which represents a less common but highly intriguing type of reasoning task.

In the few-shot version of SCoT, we conduct experiments exclusively on the MathQA, AQuA, GSM8K, and ARC datasets. This selection is due to the requirement that the dataset must have a sufficiently large training set with gold answers for constructing the demonstration corpus. We perform zero-shot SCoT on these datasets and retain only the demonstrations that produced correct answers to construct the demonstration corpus. More details on this process are provided in the Appendix A.2.

4.2 Models

To verify the effectiveness of the SCoT method, we utilize the following LLMs: the Llama3 series (Dubey et al., 2024) (including Llama3-8B, Llama3-70B, Llama3.1-8B, and Llama3.1-70B); the Llama2 series (Touvron et al., 2023) (including Llama2-7B, Llama2-13B, and Llama2-70B); Mistral-7B (Jiang et al., 2023); the Qwen2 series (Yang et al., 2024a) (including Qwen2-7B and Qwen2-72B); and ChatGLM4-9B (Team GLM et al., 2024). ChatGLM4-9B is chat-oriented and other models are instruction-tuned.

4.3 Baselines

We use zero-shot prompts (Kojima et al., 2022), Self-Consistency (Wang et al., 2022) and Step Back (Zheng et al., 2023) as baselines. Step Back is only tested on 5 datasets, as it is not suitable for others. BoT (Yang et al., 2024b) is excluded due to the unavailability of its template for reproduction.

We use accuracy as the performance metric, calculated as the average of three independent inferences for each model. Experimental parameters are provided in the appendix.

5 Experimental Results

In this section, we empirically evaluate the effectiveness of the SCoT approach. We test SCoT on two open-source models, Llama3-8B and Mistral-7B, across all datasets. To further validate its efficacy, we select one dataset from each reasoning task category and test it on all 7 models. We also assess the impact of model size, perform ablation studies on SCoT components, and conduct case studies, along with additional discussions to understand the factors influencing SCoT's effectiveness.

5.1 Results across all Datasets

The experimental results across all datasets using two models are presented in Table 1. Notably, in both vanilla zero-shot and self-consistency settings, SCoT outperforms the CoT approach in most tasks, with particularly significant improvements observed on the GSM8K dataset, where accuracy increases from 52.11% to 73.16% after incorporating strategic knowledge. Additionally, SCoT achieves a 24.13% improvement on the Tracking_Object dataset. However, the Llama3-8B model exhibits a 2.6% decrease in performance on the ARC dataset. In general, the Llama3-8B model shows an average improvement of 6.92% on all datasets, while the Mistral-7B model demonstrates an average improvement of 3.81% on comparable datasets. Compared to Step Back and CoT 0-shot+SC, SCoT also performs better than these two

Table 1: Accuracy (%) using Llama3-8B and Mistral-7B across all datasets. 'SCoT 1-shot^{-}' refers to the results obtained using the standard few-shot CoT template but with demonstrations matched by strategy and '+SC' refers to the methods using self-consistency. The highest scores for 0-shot and 1-shot are both bolded.

Model	Method	MathQA	AQuA	GSM8K	MMLU	ARC	SQA	CSQA	Object
Llama3-8B	CoT 0-shot	56.33	49.61	52.11	46.67	80.60	64.60	71.13	44.27
	CoT 0-shot+SC	57.00	51.90	48.48	49.52	**81.00**	66.00	72.06	54.00
	Step Back	56.33	50.39	–	47.78	75.80	64.64	–	–
	SCoT 0-shot	56.67	51.85	**73.16**	50.00	78.02	68.56	74.00	**68.40**
	SCoT 0-shot+SC	**60.33**	**53.94**	70.58	**52.22**	78.00	**69.00**	**75.00**	61.60
	SCoT 1-shot^{-}	56.33	50.87	74.91	–	73.40	–	–	–
	SCoT 1-shot	**57.67**	**55.12**	**76.57**	–	**80.60**	–	–	–
Mistral-7B	CoT 0-shot	30.00	29.13	36.26	29.75	67.20	56.22	61.80	21.40
	CoT 0-shot+SC	31.42	32.87	34.50	31.88	68.78	53.50	62.69	24.50
	Step Back	31.43	32.87	–	31.85	68.00	56.72	–	–
	SCoT 0-shot	30.44	33.60	**38.97**	32.35	72.20	**61.89**	**68.00**	24.75
	SCoT 0-shot+SC	**31.67**	**36.22**	34.72	**32.96**	**75.40**	57.33	66.50	**27.60**
	SCoT 1-shot^{-}	34.33	31.50	45.57	–	67.40	–	–	–
	SCoT 1-shot	**37.00**	**35.04**	**47.38**	–	**73.20**	–	–	–

Table 2: Accuracy(%) across seven models on MMLU, SQA and Tracking_Object datasets

Dataset	Method	Llama3-8B	Mistral-7b	Chatglm4-9B	Qwen2-7B	Qwen2-72B	Llama3.1-8B	Llama3.1-70B
MMLU Math	CoT 0-shot	46.67	29.75	66.67	**71.97**	84.20	**59.63**	**85.19**
	SCoT 0-shot	**50.00**$_{+3.33}$	**32.35**$_{+2.59}$	**68.15**$_{+1.48}$	71.85	**85.93**$_{+1.73}$	56.42	**85.19**
SQA	CoT 0-shot	64.60	56.22	61.80	**61.00**	75.22	73.11	64.67
	SCoT 0-shot	**68.56**$_{+3.96}$	**61.89**$_{+5.67}$	**64.67**$_{+2.87}$	**61.00**	**77.67**$_{+2.45}$	**74.22**$_{+1.11}$	**82.33**$_{+1.33}$
Object	CoT 0-shot	44.27	21.40	61.80	46.20	93.93	62.60	**100.00**
	SCoT 0-shot	**68.40**$_{+24.13}$	**24.67**$_{+3.27}$	**69.00**$_{+7.20}$	**47.53**$_{+1.33}$	**97.47**$_{+3.54}$	**77.60**$_{+15.00}$	**100.00**

methods except for the result of CoT 0-shot+SC with Llama3-8B model on the ARC dataset. Nevertheless, SCoT and SCoT+SC still achieves comparable results to it. Notably, SCoT shows substantial gains in commonsense reasoning tasks compared with other methods.

Furthermore, we extend the SCoT framework to support few-shot settings by automatically matching demonstrations, resulting in even stronger performance. The SCoT 1-shot^{-}, as shown in Table 1, refers to CoT prompting with demonstrations matched through strategic knowledge. Compared to CoT 0-shot[1], SCoT 1-shot^{-}, which uses strategy-matched demonstrations, shows significant performance improvements across most datasets, highlighting the effectiveness of the matched demonstrations. The SCoT 1-shot, which combines both strategic knowledge and strategy-matched demonstrations, achieves the best results.

5.2 Results across all Models

The experimental results for all models on the three datasets are shown in Table 2. The experiments demonstrate that SCoT can enhance performance across most models. In particular, with the exception of the Llama3.1-8B model, where the addition of SCoT results in a slight decrease in accuracy on the MMLU task, other models exhibit accuracy improvements ranging from 1.11% to 24.13% across the three datasets. Note that the CoT 0-shot has achieved 100% accuracy with Llama3.1-70B model on Tracking_Object dataset, and SCoT 0-shot maintains this performance.

5.3 Model Scale

We investigate the impact of model size on the effectiveness of SCoT, the results on the Llama2 model series with three different sizes are shown in Figure 5. It demonstrates that SCoT can lead to accuracy improvements across all sizes of the Llama2 models. However, a general trend emerges that performance improvement decreases marginally with model size. Furthermore, manual inspection of the model outputs reveals that larger models are more likely to generate CoT path containing strategic knowledge in 0-shot settings.

[1]We do not present the accuracy of CoT 1-shot separately as it was comparable to CoT 0-shot in our experiments.

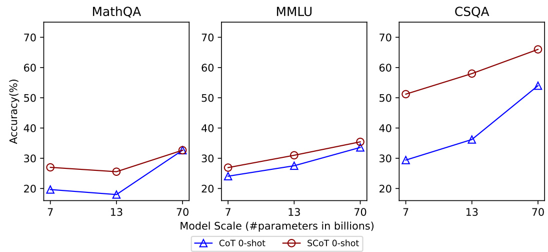


Figure 5: Accuracy(%) across three datasets using different scales of models in Llama2 series

Table 3: Ablation study on SCoT prompt components: * denotes a non-markdown format, while no * indicates a markdown format.

Method	AQuA	ARC
Mistral-7B*	29.13%	67.20%
Mistral-7B + Role*	27.95%	69.80%
Mistral-7B + Role	32.28%	71.20%
Mistral-7B + WorkFlow*	33.07%	70.40%
Mistral-7B + WorkFlow	31.89%	70.40%
SCoT 0-shot (Ours)	33.60%	72.20%
SCoT 1-shot (Ours)	35.04%	**73.20%**
SCoT 3-shot (Ours)	**35.43%**	**73.20%**

5.4 Ablation Study

We explore the effects of various components within the prompt (such as role, workflow, structure, and the quantity of demonstrations) on accuracy. The experimental results are illustrated in Table 3. Building on the CoT 0-shot approach, we observed that adding roles, incorporating workflows, and formatting prompts in markdown progressively increased accuracy. We also explored the impact of the number of demonstrations on accuracy within the few-shot SCoT framework. Experimental results indicate that as the number of demonstrations increases, the performance of SCoT either slightly improves or remains unchanged.

5.5 Case Study

We conduct a detailed case study focusing on the validity of the strategic knowledge elicited from the model. Figure 6 shows several typical cases.

In the domain of mathematics, we observe that the SCoT output tends to favor solving problems using inequalities rather than directly analyzing the problem to reach an answer. For the instance of frog jumping calculation in the Figure 6, an incorrect solution may miscalculate the final jump's impact. While generating a strategy ensures accurate calculations by considering all constraints and systematically solving the problem.

In physical tasks, we find that the model's CoT output could be misled by specific phrases in the task input (*e.g.*, "capacitor"), leading to the selection of an incorrect formula. In contrast, the SCoT approach successfully elicited the correct formula. Similarly, in multi-hop reasoning tasks, CoT output often focuses on details, resulting in incomplete subsequent logical reasoning, whereas SCoT generates answers by considering the overall context.

Table 4: Token length comparison for SCoT and CoT 0-shot methods

Dataset	Method	Llama3-8B	Mistral-7B
AQuA	CoT 0-shot	361.384	270.260
	SCoT 0-shot	**370.378**	**458.413**
GSM8K	CoT 0-shot	130.532	**858.507**
	SCoT 0-shot	**206.278**	611.848
Object	CoT 0-shot	121.460	89.654
	SCoT 0-shot	**174.888**	**162.822**

5.6 Efficiency Analysis

Due to SCoT's mechanism of generating strategy before solving problems in one query, it is more efficient than multi-query methods. However, compared to single-query methods, the output token length might be longer, potentially decreasing efficiency. To investigate this, we measure the token lengths for the AQuA, GSM8K, and Tracking_Object using both CoT 0-shot and SCoT 0-shot methods. The results are shown in Table 4.

The results indicate that the token length output by the Mistral-7B model on the GSM8K dataset decreases with SCoT. This reduction may be due to the model's tendency to repetitively generate a specific answer span up to the inference length limit on the GSM8K dataset in CoT 0-shot, leading to a decline in accuracy. SCoT mitigates this issue. Besides, the length of SCoT varies from 1.03 to 1.8 times that of CoT, averaging around 1.5 times. This shows that while our method is somewhat slower than CoT, the efficiency remains manageable.

6 Discussions

6.1 Automatic SCoT

To demonstrate that our experimental results are not influenced by human-crafted prompts but rather due to the concept of SCoT, we conduct a preliminary test to evaluate whether the SCoT prompt templates can be automatically generated. We provide the SCoT concept to LLMs to generate the

Math Question A	A well is 30 meters deep. A frog at the bottom jumps 3 meters up each day but slips back 2 meters each night. How many days will it take for the frog to climb out of the well?
CoT Answer	To solve this problem, we need to **consider the frog's daily progress**. On the first day, the frog jumps up 3 meters but slips back 2 meters, resulting in a net gain of 1 meter. From the second day onward, the frog gains 1 meter each day. The frog needs to climb a total of 30 meters to get out of the well. After the first day, it still needs to climb 29 meters. Therefore, the total number of days required is **1+29=30 days**.
SCoT Answer	To address this problem elegantly, we can **utilize a mathematical model**. The frog gains a net height of 1 meter each day. However, on the day it reaches the top of the well, it will no longer slip back down. Let n represent the number of days it takes for the frog to climb out of the well. We can **formulate the following inequality**: n - 1 + 3 >= 30 This simplifies to: n >= 28 Thus, the total number of days for the frog to climb out of the well is **28 days**.
Physical Question B	How much work is required to charge a 10μF capacitor to a potential difference of 100V?
CoT Answer	Using the formula for work done by a capacitor, which is **W=k×V×C**, and substituting the values, we find that **0.09J** of work needs to be done.
SCoT Answer	Using the capacitor charging formula. The formula for capacitor charging is **E = (1/2)CV^2** To substitute the values, it needs to do **0.05J** of work
Multi-hop Question C	Is shrimp scampi definitely free of plastic?
CoT Answer	Shrimp scampi is a dish made with shrimp, butter, garlic, and sometimes lemon, usually served with pasta. The ingredients typically include fresh or frozen shrimp, butter, garlic, and lemon juice. **There is no connection between shrimp scampi and plastic, as plastic is not an ingredient in this dish**. Therefore, shrimp scampi is **True** definitely free of plastic.
SCoT Answer	Shrimp scampi usually does not contain plastic, but contamination is possible. It is typically made with ingredients like shr imp, garlic, butter, and parsley, which do not contain plastic, and **there are no known instances of shrimp scampi being made with plastic-containing ingredients**. Therefore, while shrimp scampi is usually made without plastic. **So, the answer is: No**.

Figure 6: Comparison of the paths generated by CoT and SCoT in different domains using the Qwen2-72B model.

Table 5: Accuracy(%) using automatically generated prompts based on the SCoT concept using Mistral-7b

Method	Accuracy
CoT 0-shot	29.13
SCoT 0-shot	33.60
Auto SCoT	**31.89**

Table 6: Accuracy (%) using DeepSeek-R1-Distill-Qwen-32B.

	MathQA	MMLU	ARC	Object
CoT 0-shot	55.67	85.56	**95.00**	99.70
SCoT 0-shot	**84.33**	**93.70**	94.40	**99.80**

corresponding prompt templates and tested these on the AQuA dataset. The results are presented in Table 5. The findings indicate that while the accuracy of prompts automatically generated based on the SCoT concept is lower than that of manually crafted SCoT prompts, it is still superior to 0-shot CoT performance. This suggests that the automatic generation of SCoT-based prompt is feasible.

6.2 Enhancing Smaller-Scale Models

In this paper, we focus on enhancing the reasoning capabilities of smaller-scale models. We exclude larger, more powerful models from our experiments because they already achieve accuracy rates exceeding 95% on our datasets, even in a zero-shot CoT configuration. This indicates that the capabilities of models extremely large parameters on these tasks are already saturated. In future work, we aim to test SCoT on more challenging reasoning datasets to further validate its efficacy on stronger models.

6.3 Accuracy of the DeepSeek Distilled Model

We also experimented with the popular DeepSeek-R1 series models (DeepSeek-AI et al., 2025), specifically usingCoT 0-shot and SCoT 0-shot on the DeepSeek-R1-Distill-Qwen-32B model across four datasets, as shown in Table 6. Except for a slight negative gain on the ARC dataset, we observed positive gains on the other three datasets, with the highest improvement of 28.66% on MathQA. These results are consistent with those in Table 1, demonstrating that SCoT is effective on the DeepSeek series models.

7 Conclusion

In this paper, we introduce the Strategic Chain-of-Thought, a method that enables large models to autonomously generate an optimal Chain-of-Thought path. By integrating a structured workflow for eliciting and applying strategic knowledge, SCoT enhances the model's ability to produce a high quality outputs. We further extend SCoT to a few-shot version by matching demonstrations through strategic knowledge from a predefined strategic knowledge-based corpus. Experimental results demonstrate the effectiveness of both 0-shot and few-shot SCoT.

Overall, SCoT offers a promising framework for improving the quality of reasoning path in large models. Future research will focus on evaluating its effectiveness with more complex problems and exploring further applications.

8 Limitation

In this paper, we propose a novel methodology to enhance the performance of large language models by incorporating strategic knowledge prior to generating intermediate reasoning steps. However, there are three limitations in our work.

The first limitation is that we only use one-shot SCoT in our main experiments. This choice was made because incorporating additional demonstrations would significantly increase input length, leading to substantial resource consumption. As a result, we limited our experiments to a single dataset and model without conducting large-scale testing.

The second limitation is the exclusion of several recent methods from our baseline comparisons. For example, approaches like Buffer-of-Thought could not be reproduced due to the lack of clear implementation details in the original paper. Other methods, such as Plan-and-Solve, were excluded as they showed suboptimal performance in our preliminary experiments.

The third limitation is that our experiments focused solely on reasoning tasks, leaving us uncertain about SCoT's effectiveness in other domains. Additionally, we have not provided a theoretical proof to explain why SCoT is effective.

We plan to address these limitations in future work by expanding the experimental scope and refining our methodology.

9 Ethical Considerations

All datasets and models used in this paper are open-source, and the licenses for the models have been specified. The prompts used in the experiments are provided, and the entire study can be reproduced using widely available large model API frameworks. This ensures the reproducibility and transparency of the research.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *Preprint*, arXiv:1905.13319.

BIG-bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

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. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Preprint*, arXiv:2101.02235.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Jie Huang and Kevin Chen-Chuan Chang. 2022. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, and et al. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

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.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113, Mexico City, Mexico. Association for Computational Linguistics.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation : Learning to solve and explain algebraic word problems. *Preprint*, arXiv:1705.04146.

Richard Rusczyk. 2003. *The Art of Problem Solving*. Mathematical Association of America, Washington, D.C.

Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. 2023. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing. *Preprint*, arXiv:2309.08008.

Mirac Suzgun and Adam Tauman Kalai. 2024. Meta-prompting: Enhancing language models with task-agnostic scaffolding. *arXiv preprint arXiv:2401.12954*.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive Science*, 12(2):257–285.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, and et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, and et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

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

Xuezhi Wang and Denny Zhou. 2024. Chain-of-thought reasoning without prompting. *arXiv preprint arXiv:2402.10200*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, and et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. *ArXiv*, abs/2309.03409.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024b. Buffer of thoughts: Thought-augmented reasoning with large language models. *arXiv preprint arXiv:2406.04271*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2023. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. 2023. Take a step back: Evoking reasoning via abstraction in large language models. *arXiv preprint arXiv:2310.06117*.

A Details of Experiments

A.1 Models Details

This experiment involves eleven models, nine of which are public (Llama3-8B, Llama2-7B, Mistral-7B, Llama3.1-8B, Qwen2-7B, ChatGLM4-9B, Llama3-70B, Llama3.1-70B, Llama2-70B, Qwen2-72B and Qwen2.5-72B). The sources and licenses for all public models are detailed in Table 7.

A.2 Datasets Details

This experiment involves eight datasets: MathQA, AQuA, GSM8K, MMLU, ARC, StrategyQA, CommonsenseQA, and Tracking_Object. Due to the large size of the MathQA, StrategyQA, ARC, CommonsenseQA, and Tracking_Object datasets, we randomly selected a subset from each to serve as the test dataset, while for the other datasets, we used the original full dataset. All datasets used in this study are publicly available, with their sources, licenses and size detailed in Table 8.

MathQA, AQuA, MMLU, ARC, StrategyQA, CommonsenseQA, and Tracking_Object consist of multiple-choice questions. To determine correctness, we compare the predicted choice with the gold (correct) choice. For GSM8K, the answers are numerical text spans; we assess correctness by checking if the predicted answer exactly matches the gold answer.

As mentioned in Section 3, we exclusively conducted experiments on the MathQA, AQuA, GSM8K, and ARC datasets, as these were the only four datasets in our study that contained training sets. Moreover, the demonstration corpus for the few-shot version of SCoT required these training sets to provide demonstrations that differed from those in the test set and aligned appropriately with it. Consequently, we limited our experiments to these four datasets.

Initially, we executed the zero-shot version of SCoT using the training sets and subsequently evaluated the final results as the gold answers are provided in the training sets. We assumed that if the model correctly answered a question, the strategic knowledge it generated was also accurate. Therefore, we retained only the correct demonstrations to construct the demonstration corpus for the few-shot version of SCoT. This systematic approach ensured the integrity of our evaluation and enhanced the overall effectiveness of the few-shot learning framework.

Table 7: Models, sources and licenses used in this work

Models	Modelsources	License
Llama2-7B-chat	https://huggingface.co/meta-llama/Llama-2-7b-chat	llama2 license
Llama2-13B	https://huggingface.co/meta-llama/Llama-2-13b-chat	llama2 license
Llama2-70B	https://huggingface.co/meta-llama/Llama-2-70b-chat	llama2 license
Llama3-8B	https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct	llama3 license
Llama3.1-8B	https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct	llama3.1 license
Llama3.1-70B	https://huggingface.co/meta-llama/Meta-Llama-3.1-70B-Instruct	llama3.1 license
Mistral-7B	https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2	tongyi-qianwen license
Qwen2-7B	https://huggingface.co/Qwen/Qwen2-7B-Instruct	tongyi-qianwen license
Qwen2-72B	https://huggingface.co/Qwen/Qwen2-72B-Instruct	Apache License 2.0
Qwen2.5-72B	https://huggingface.co/Qwen/Qwen2.5-72B-Instruct	Qwen license
ChatGLM4-9b	https://huggingface.co/THUDM/glm-4-9b-chat	glm-4-9b License
DeepSeek-R1-Distill-Qwen-32B	https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B	MIT License

Table 8: Datasets, sources and licenses used in this work

Datasets	Sources	Licenses	Size
MathQA	https://huggingface.co/datasets/datafreak/MathQA	Apache License 2.0	300
AQuA	https://github.com/google-deepmind/AQuA	Apache License 2.0	254
GSM8K	https://huggingface.co/datasets/openai/gsm8k	MIT License	1319
MMLU	https://huggingface.co/datasets/cais/mmlu	MIT License	300
ARC	https://huggingface.co/datasets/allenai/ai2_arc	CC-BY-SA-4.0 License	270
StrategyQA	https://huggingface.co/datasets/ChilleD/StrategyQA/viewer/default/test	MIT License	500
CommonsenseQA	https://huggingface.co/datasets/tau/commonsense_qa	MIT License	500
Object Tracking	https://github.com/google/BIG-bench/tree/092b196c1f8f14a54bbc62f24759d43-bde46dd3b/bigbench/benchmark_tasks/tracking_shuffled_objects/three_objects	Apache License 2.0	200

A.3 Other Details

We used the standard zero-shot CoT and few-shot CoT templates, with the Step Back template following the design from the original paper, and the parameters for self-consistency also based on the original work. Additionally, the selection of the normal one-shot demonstration was done using embedding and cosine similarity, similar to the approach in RAG.

For all experiments, except those involving Self-Consistency, the temperature is set to 0, and the top_p parameter is set to 1. For Self-Consistency, following the settings from the original paper (Wang et al., 2022), the temperature is adjusted to 0.5, and top_p is set to 0.5. 20 responses are selected for voting with -SC method.

We utilize vllm (Kwon et al., 2023) as the inference framework for all deployments. For models under 70 billion parameters (such as Llama3-8B, Llama2-7B, and Mistral-7B), we deploy each on a single 32GB AI graphics card. For models with 70 billion parameters or more (including Llama3-70B, Llama3.1-70B, Llama2-70B, and Qwen2-72B), we utilize two 80GB AI graphics cards per model.

B Results

B.1 All Results

Accuracy is used as the evaluation metric. We conducted three independent inference runs for all experiments and calculated the average results. However, due to the high computational cost, we performed only a single inference for Self-Consistency. The accuracy and standard deviation results are presented in Table 9 and Table 10.

Table 9: Accuracy (%) using Llama2-8B and Mistral-7B across all datasets. SCoT 1-shot^{-} refers to the results obtained using the standard few-shot CoT template but with demonstrations matched by strategy.

Model	Method	MathQA	AQuA	GSM8K	MMLU	ARC	SQA	CSQA	Object
Llama3-8B	CoT 0-shot	$56.33_{\pm 0.000}$	$49.61_{\pm 1.790}$	$52.11_{\pm 0.129}$	$46.67_{\pm 0.000}$	$80.60_{\pm 0.000}$	$64.60_{\pm 0.646}$	$71.13_{\pm 0.094}$	$44.27_{\pm 0.736}$
	CoT 0-shot+SC	57.00	51.90	48.48	49.52	**81.00**	66.00	72.06	54.00
	Step Back	$56.33_{\pm 0.272}$	$50.39_{\pm 0.000}$	–	$47.78_{\pm 0.000}$	$75.80_{\pm 0.248}$	$64.64_{\pm 0.2722}$	–	–
	SCoT 0-shot	$56.67_{\pm 0.000}$	$51.85_{\pm 1.299}$	**73.16**$_{\pm 0.163}$	$50.00_{\pm 0.000}$	$78.02_{\pm 0.000}$	$68.56_{\pm 0.566}$	$74.00_{\pm 0.000}$	**68.40**$_{\pm 0.000}$
	SCoT 0-shot+SC	**60.33**	**53.94**	70.58	**52.22**	78.00	**69.00**	**75.00**	61.60
	SCoT 1-shot^{-}	$56.33_{\pm 0.000}$	$50.87_{\pm 2.140}$	$74.91_{\pm 0.000}$	–	$73.40_{\pm 0.000}$	–	–	–
	SCoT 1-shot	**57.67**$_{\pm 0.000}$	**55.12**$_{\pm 0.000}$	**76.57**$_{\pm 0.000}$	–	**80.60**$_{\pm 0.000}$	–	–	–
Mistral-7B	CoT 0-shot	$30.00_{\pm 0.000}$	$29.13_{\pm 1.245}$	$36.26_{\pm 1.854}$	$29.75_{\pm 0.924}$	$67.20_{\pm 0.356}$	$56.22_{\pm 0.314}$	$61.80_{\pm 0.000}$	$21.40_{\pm 0.000}$
	CoT 0-shot+SC	31.42	32.87	34.50	31.88	68.78	53.50	62.69	24.50
	Step Back	$31.43_{\pm 0.000}$	$32.87_{\pm 0.322}$	–	$31.85_{\pm 0.495}$	$68.00_{\pm 0.000}$	$56.72_{\pm 0.000}$	–	–
	SCoT 0-shot	$30.44_{\pm 0.874}$	$33.60_{\pm 1.523}$	**38.97**$_{\pm 0.655}$	$32.35_{\pm 1.665}$	$72.20_{\pm 0.370}$	**61.89**$_{\pm 0.415}$	**68.00**$_{\pm 0.000}$	$24.75_{\pm 0.165}$
	SCoT 0-shot+SC	**31.67**	**36.22**	34.72	**32.96**	**75.40**	57.33	66.50	**27.60**
	SCoT 1-shot^{-}	$34.33_{\pm 0.000}$	$31.50_{\pm 0.964}$	$45.57_{\pm 1.087}$	–	$67.40_{\pm 0.000}$	–	–	–
	SCoT 1-shot	**37.00**$_{\pm 0.000}$	**35.04**$_{\pm 0.000}$	**47.38**$_{\pm 0.107}$	–	**73.20**$_{\pm 0.000}$	–	–	–

Table 10: Accuracy(%) across seven models on MMLU, SQA and Tracking_Object datasets

Dataset	Method	Llama3-8B	Mistral-7b	Chatglm4-9B	Qwen2-7B	Qwen2-70B	Llama3.1-8B	Llama3.1-70B
MMLU Math	CoT 0-shot	$46.67_{\pm 0.000}$	$29.75_{\pm 0.924}$	$66.67_{\pm 0.302}$	**71.97**$_{\pm 0.349}$	$84.20_{\pm 0.349}$	**59.63**$_{\pm 0.000}$	**85.19**$_{\pm 0.605}$
	SCoT 0-shot	**50.00**$_{\pm 0.000}$	**32.35**$_{\pm 1.665}$	**68.15**$_{\pm 0.907}$	$71.85_{\pm 0.302}$	**85.93**$_{\pm 0.302}$	$56.42_{\pm 0.175}$	**85.19**$_{\pm 0.000}$
SQA	CoT 0-shot	$64.60_{\pm 0.595}$	$56.22_{\pm 0.314}$	$61.80_{\pm 0.363}$	**61.00**$_{\pm 0.000}$	$75.22_{\pm 0.314}$	$73.11_{\pm 0.314}$	$64.67_{\pm 0.000}$
	SCoT 0-shot	**68.56**$_{\pm 0.566}$	**61.89**$_{\pm 0.415}$	**64.67**$_{\pm 0.408}$	**61.00**$_{\pm 0.157}$	**77.67**$_{\pm 0.272}$	**74.22**$_{\pm 0.157}$	**82.33**$_{\pm 0.157}$
Object	CoT 0-shot	$44.27_{\pm 0.736}$	$21.40_{\pm 0.000}$	$61.80_{\pm 0.000}$	$46.20_{\pm 0.000}$	$93.93_{\pm 0.525}$	$62.60_{\pm 0.411}$	**100.00**$_{\pm 0.000}$
	SCoT 0-shot	**68.40**$_{\pm 0.000}$	**24.67**$_{\pm 0.000}$	**69.00**$_{\pm 0.000}$	**47.53**$_{\pm 0.094}$	**97.47**$_{\pm 0.339}$	**77.60**$_{\pm 0.993}$	**100.00**$_{\pm 0.000}$

B.2 Case Study

We conducted a detailed case study to assess the validity of the strategic knowledge elicited from the model. Figures 7 and 8 present several representative cases spanning math reasoning, physical reasoning, commonsense reasoning, multi-hop reasoning, and spatial reasoning.

C Comparison with Step Back

We have presented the results of Step Back in Table 1. The results show that Step Back prompting performs comparably to zero-shot CoT, with some results even falling below the baseline.

The motivation behind Step Back is that the original question may lead the model to focus excessively on details. By abstracting the original question, it becomes easier to align the model with relevent knowledge, guiding its reasoning path. The method involves generating principles or using the Step Back question for RAG. However, the principles or results from RAG are distinct from strategic knowledge. The abstraction results do not always correspond to strategic knowledge, and an abstracted principle does not always provide effective guidance for problem-solving.

In STEM tasks, Step Back first generates a principle. For other tasks, a Step Back question is first generated, and then the retrieval results from both the original and Step Back questions are combined to generate the final answer.

For STEM tasks, the principles abstracted from a problem do not necessarily guide the model toward a correct solution. After analyzing the experimental results of Step Back, we conclude that the primary issue is that many math problems lack an overarching principle. For example, consider the problem: "The price of a product is reduced by 30%. By what percentage should it be increased to make it 100%?" This problem can be directly solved by setting up an equation. SCoT would generate the equation:

In contrast, Step Back fails to find a formula and outputs incorrect reasoning steps:

"The price is reduced by 30%, so the new price is 70% of the original price. To make it 100%, we need to increase the price by: 100% - 70% = 30%. Thus, the price should be increased by 30% to make it 100%."

For other tasks, abstracting the original question does not always lead to better retrieval of relevant knowledge. For example, consider the question: "Are more people today related to Genghis Khan than Julius Caesar?" The model generates the Step

Math Question A	A well is 30 meters deep. A frog at the bottom jumps 3 meters up each day but slips back 2 meters each night. How many days w ill it take for the frog to climb out of the well?
CoT Answer	To solve this problem, we need to **consider the frog's daily progress**. On the first day, the frog jumps up 3 meters but slips back 2 meters, resulting in a net gain of 1 meter. From the second day onward, the frog gains 1 meter each day. The frog needs to climb a total of 30 meters to get out of the well. After the first day, it still needs to climb 29 meters. Therefore, the total number of days required is **1+29=30 days**.
SCoT Answer	To address this problem elegantly, we can **utilize a mathematical model**. The frog gains a net height of 1 meter each day. However, on the day it reaches the top of the well, it will no longer slip back down. Let n represent the number of days it takes for the frog to climb out of the well. We can **formulate the following inequality**: n - 1 + 3 >= 30 This simplifies to: n >= 28. Thus, the total number of days for the frog to climb out of the well is **28 days**.
Physical Question B	How much work is required to charge a 10μF capacitor to a potential difference of 100V?
CoT Answer	Using the formula for work done by a capacitor, which is **W=k×V×C**, and substituting the values, we find that **0.09J** of work needs to be done.
SCoT Answer	Using the capacitor charging formula. The formula for capacitor charging is **E = (1/2)CV^2** To substitute the values, it needs to do **0.05J** of work
Math Question C	Jordan plays video games for 2 hours every day. He also has a part-time job where he earns $10 an hour. How much money would Jordan earn in one week if he spent his video game time working instead?
CoT Answer	To calculate how much money Jordan would earn in one week if he spent his video game time working instead, we need to follow these steps: Step 1: Jordan plays video games for 2 hours every day. There are 7 days in a week, so he would work for 7 - 2 = 5 hours each day. Step 2: Jordan earns 10 an hour. If he works 5 hours a day, his daily earnings would be 5 hours * 10/hour = 50. To find his weekly earnings, we multiply his daily earnings by the number of days in a week: 50/day * 7 days/week = **350/week.**
SCoT Answer	Calculate the total hours he would have worked in a week by multiplying the number of hours he plays per day by the number of days in a week. Given: 1. Hours Jordan plays video games per day = 2 hours; 2. Days in a week = 7 days Total hours = 2 hours/day ×7 days/week = 14 hours/week Calculate the total amount of money Jordan would have earned in one week if he worked those 14 hours. Total earnings = 14 hours/week ×$10/hour = **$140/week**

Figure 7: Comparison of the paths generated by CoT and SCoT in different domains.

Back question: "Who are the ancestors of a large number of people today?" Similarly, for the question "Do the anchors on Rede Globo speak Chinese?" the Step Back question becomes: "What languages are commonly spoken by the anchors on Rede Globo?" While these Step Back questions are related to the originals, solving them does not necessarily help answer the original questions and, in some cases, could mislead the model. For instance, in the second question, anchors on Rede Globo may commonly speak Brazilian Portuguese, but this has no relevance to whether they speak Chinese. Including such information could have a negative impact on the model's answer. Thus, while abstracting a problem can help resolve some reasoning tasks, it is not always effective in all cases. In contrast, SCoT performs well in these tasks.

Additionally, Step Back involves a multi-step prompting process (two steps for STEM tasks and three steps for other tasks), whereas SCoT requires only one step, making it more efficient. Experimental results also indicate that Step Back shows limited effectiveness on smaller models when using multiple steps. Manual analysis revealed that smaller models often fail to generate correct knowledge in the first step. In some cases, they generate irrelevant content or produce answers directly rather than the necessary knowledge. This failure to generate accurate knowledge compounds in subsequent steps, leading to increasingly erroneous outputs.

D Experimental Prompts

The prompt for standard zero-shot Chain-of-Thought is shown in Figure 9. Prompts for zero-shot Strategic Chain-of-Thought are displayed in Figure 10 (for math reasoning), Figure 11 (for multi-hop reasoning), Figure 13 (for physical reasoning), Figure 12 (for commonsense reasoning) and Figure 14 (for spatial reasoning). Prompts for one-shot Strategic Chain-of-Thought are shown in Figure 15. The prompt for one-shot Strategic Chain-of-Thought is provided in Figure 5. Finally, the prompts for automated Strategic Chain-of-Thought are shown in Figure 6. The automated SCOT prompts were generated using the Qwen2.5-72B model by given the idea of SCoT. We did not provide pseudocode directly in the paper; instead, we offered prompts, as utilizing these prompts along with standard API call codes allows for straightforward reproduction of our work.

Object Question A	Alice, Bob, and Claire are playing a game. At the start of the game, they are each holding a ball: Alice has a brown ball, Bo b has a red ball, and Claire has a purple ball. As the game progresses, pairs of players trade balls. First, Bob and Claire swap balls. T hen, Alice and Claire swap balls. Finally, Alice and Bob swap balls. At the end of the game, Claire has the A) brown ball; B) red ball; C) purple ball
CoT Answer	Let's break down the sequence of swaps: 1. Bob and Claire swap balls: Bob gets the purple ball, and Claire gets the red ball. 2. Alice and Claire swap balls: Alice gets the red ball (which Claire had just gotten), and Claire gets the brown ball. 3. Alice and Bob swap balls: Alice gets the red ball (which she had just given to Claire), and Bob gets the brown ball. So, at the end of the game, the final positions are: Claire has **the purple ball.**
SCoT Answer	Initial State: Alice has a brown ball; Bob has a red ball; Claire has a purple ball Process Exchanges: 1. Bob and Claire swap balls: Alice still has a brown ball; Bob has a purple ball; Claire has a red ball 2. Alice and Claire swap balls: Alice has a red ball; Bob has a purple ball; Claire has a brown ball 3. Alice and Bob swap balls: Alice has a purple ball; Bob has a red ball; Claire has a brown ball Determine the Answer: The correct answer is that Claire has the **brown ball.**
Math Question B	Find out which of the following values is the multiple of X, if it is divisible by 9 and 12? A) 36; B) 15; C) 17; D) 5; E) 7
CoT Answer	To find the multiple of a number that is divisible by both 9 and 12, we need to find the common multiple of 9 and 12. The least common multiple (LCM) of 9 and 12 can be found using the prime factorization method or the LCM table. The prime factorization of 9 is 3^2, and the prime factorization of 12 is 2^2 * 2 * 3. Since 3^2 is a factor of both 3^2 and 2^2 * 2 * 3, the LCM of 9 and 12 is 3^2 * 2^2 * 2 * 3 = 3^2 * 2^2 * 3 = 3^2 * 9 = 9^2 = 81. Now, let's check which option is a perfect square of 9. Option **B, 15**, is a perfect square of 9, it is the correct answer..
SCoT Answer	The problem states that X is divisible by 9 and 12. To find the multiple of X that fits the given conditions, we need to find the least common multiple (LCM) of 9 and 12. Choose the most efficient and practical approach: the LCM of 9 and 12 is 36. Therefore, the correct answer **is A) 36.**
Multi-hop Question C	Is shrimp scampi definitely free of plastic?
CoT Answer	Shrimp scampi is a dish made with shrimp, butter, garlic, and sometimes lemon, usually served with pasta. The ingredients typ ically include fresh or frozen shrimp, butter, garlic, and lemon juice. **There is no connection between shrimp scampi and plastic, as plastic is not an ingredient in this dish**. Therefore, shrimp scampi is **True** definitely free of plastic.
SCoT Answer	Shrimp scampi usually does not contain plastic, but contamination is possible. It is typically made with ingredients like shr imp, garlic, butter, and parsley, which do not contain plastic, and **there are no known instances of shrimp scampi being made with plastic-containing ingredients**. Therefore, while shrimp scampi is usually made without plastic. **So, the answer is: No.**

Figure 8: Comparison of the paths generated by CoT and SCoT in different domains.

Zero-shot CoT template	I will provide you with a math problem and 5 options. Please choose the correct option from the five provided and indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 9: An example of prompting for standard zero-shot CoT

Zero-shot SCoT template	# Role A highly skilled mathematician and algorithm expert. # Workflow 1. Analyze the problem and identify any relevant mathematical formulas, or approaches that might be helpful, and select the approaches that can solve the problem. 2. Choose the most efficient and practical approach. For example, when asked to find the sum of all integers from -25 to 23, consider using the summation formula of arithmetic sequence instead of simply adding the numbers one by one. The summation formula of arithmetic sequence is an elegant and practical solution, while rudely adding the numbers is not. 3. Solve the problem step by step following the selected approach carefully. ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the math problem following <Workflow>.I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 10: An example of prompting for standard Strategic Chain-of-Thought in math reasoning tasks

Zero-shot SCoT template	# Role An expert of world knowledge with strong logical skills. # Workflow 1. Analyze the problem and break down the complex query into simpler sub-questions. 2. Sequentially finding reliable answers for each sub-question. 3. Integrating these answers to form a comprehensive. Directly answering the main question is rude, but breaking it down, answering the sub-questions, and then integrating the answers is elegant and practical. ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the problem following <Workflow>.I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 11: An example of prompting for standard Strategic Chain-of-Thought in multi-hop reasoning tasks

Zero-shot SCoT template	# Role An expert with world knowledge and reasoning abilities. # Workflow 1. Understanding the Question: Identify key concepts and comprehend the question's context. Ensure you grasp the main idea and any analogies being used. Search for any concept, knowledge, or common sense related to the topic. 2. Analyzing the Options: Read each choice carefully, understand its meaning, and relate it to the question's context to determine relevance. 3. Logical Reasoning: Use logical reasoning to eliminate options that are clearly irrelevant or incorrect based on the question's context. Compare the remaining options to identify the one that best aligns with the question's requirements and the context provided. ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the problem following <Workflow>.I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 12: An example of prompting for standard Strategic Chain-of-Thought in commonsense reasoning tasks

Zero-shot SCoT template	# Role A careful expert proficient in various world knowledge. # Workflow 1. Careful Question Analysis: - Read the Problem and the Options Carefully: Ensure you understand the background and specific question being asked . - Identify Keywords: Extract key terms or phrases from the Problem and the Options, try recalling their meanings . - Understand the Problem: Ensure you clearly understand what the Problem is asking, including any specific conditions or requirements. Eliminate options that are not relevant to the problem. 2. Identify Relevant Knowledge and approaches: - Recall Related Knowledge or approach: Identify all the relevant concepts, principles, or formulas that might apply to the Problem. - Select Appropriate Knowledge: Choose the knowledge, formulas and approaches that can solve the problem. 3. Choose the Most Efficient and Practical Knowledge and Formulas: When solving the problem, select the most efficient and practical knowledge, formulas or approaches. For example, when the description of a problem is related to potential energy and kinetic energy of an object, after using the formula PE = mgh, carefully analyze each option to judge right or wrong, rather than relying on experience or ready-made theorems to select options. 4. Careful Application of Knowledge and Formulas: -Detailed Analysis: When applying formulas and knowledge, pay attention to the specific conditions and variables in the problem. - Logical Reasoning: Carefully analyze each variable in the formula or methodically derive conclusions based on the knowledge point, ensuring the reasoning process is consistent and correct. For example, when using PE = mgh, you need to analyze the overall effect of all variables, including m, g, and h, rather than just one variable. ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the problem following <Workflow>.I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 13: An example of prompting for standard Strategic Chain-of-Thought in physical reasoning tasks

Zero-shot SCoT template	# Role A very meticulous logical Analyst. # Workflow 1. Initial State: First, list the initial state of the balls each person has according to the problem statement. 2. Process Exchanges: Next, carefully read the problem statement. For each exchange, update the current state of the balls and document the result of each exchange. 3. Determine the Answer: Once all exchanges are completed, identify which friend's ball color is being inquired about in the problem statement and select the correct answer. ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the problem following <Workflow>.I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 14: An example of prompting for standard Strategic Chain-of-Thought in spatial reasoning tasks

One-shot SCoT template	# Role A highly skilled mathematician and algorithm expert. # Workflow 1. Analyze the problem and identify any relevant mathematical formulas, or approaches that might be helpful, and select the approaches that can solve the problem. 2. Choose the most efficient and practical approach. For example, when asked to find the sum of all integers from -25 to 23, consider using the summation formula of arithmetic sequence instead of simply adding the numbers one by one. The summation formula of arithmetic sequence is an elegant and practical solution, while rudely adding the numbers is not. 3. Solve the problem step by step following the selected approach carefully. ## Demonstrations Problem: [Please Put Your Demonstration Problem Here] Options: [Please Put Your Demonstration Options Here] Answer: [Please Put Your Demonstration Answer Here] ## Rules 1. Avoid using brute force methods, as they do not reflect the professionalism. 2. Indicate your answer with [Answer]option[Answer], such as [Answer]C[Answer]. Please output the answer at the end in strict accordance with the output format. ## Initialization As <Role>, please follow <Rules> strictly. Your task is to solve the math problem following <Workflow>, <Demonstration> is some examples. I will provide you with a problem and 5 options. Please choose the correct option from the five provided. Problem: [Please Put Your Questions Here] Options: [Please Put Your Options Here] Answer: Let's think step by step.

Figure 15: An example of prompting for one-shot Strategic Chain-of-Thought