Self-Guide: Enhancing LLM Reasoning Ability via Self-Plan

Yibin Liu[1], **Zhenghao Liu**[1*], **Yukun Yan**[2], **Shi Yu**[2], **Shuo Wang**[2],
Liner Yang[3], **Huimin Chen**[4], **Yu Gu**[1], **Ge Yu**[1]
[1]Department of Computer Science and Technology, Northeastern University, Shenyang, Liaoning
[2]Department of Computer Science and Technology, Tsinghua University, Beijing
[3]School of Information Science, Beijing Language and Culture University, Beijing
[4]School of Journalism and Communication, Tsinghua University, Beijing
liuyibin@stumail.neu.edu.cn, {liuzhenghao, guyu, yuge}@mail.neu.edu.cn
{yanyk.thu, lineryang, wangshuo.thu}@gmail.com
{yus21, huimchen}@mails.tsinghua.edu.cn

Abstract

Despite significant advancements of Large language Models (LLMs) in NLP tasks, they still face the cognitive overload problem, especially in domains requiring complex reasoning. Specifically, during reasoning, LLMs need to process and memorize vast amounts of information during the reasoning process. Thus, it is a pressing issue to effectively reduce the cognitive load during the reasoning process of LLM to alleviate potential cognitive overload. We introduce the Self-Guide method to alleviate the problem, which boosts LLMs' reasoning abilities by leveraging self-generated common sense knowledge and reasoning instructions. Experimental results demonstrate that our Self-Guide method outperforms baseline methods significantly on four common reasoning tasks, which effectively enhances the reasoning ability of LLMs.

1 Introduction

In the field of Natural Language Processing (NLP) and related domains, large language models such as Chat-GPT, ChatGLM, MiniCPM, ERNIE, and Qwen have made significant advancements Brown et al. [2020], Du et al. [2021], Hu et al. [2024], Sun et al. [2021], Bai et al. [2023]. Research has shown that as the scale of model parameters rapidly increases, language models exhibit an emergent phenomenon, demonstrating their effectiveness across many NLP tasks and displaying stronger capabilities in reasoning and planning tasks Wei et al. [2022a]. However, despite their outstanding performance in many tasks, large language models still face significant challenges in multi-step reasoning required to solve complex problems Valmeekam et al. [2022], Huang and Chang [2022].

Traditional Chain-of-Thought (CoT) based reasoning methods have shown good results in many complex reasoning tasks such as mathematical calculations and relational reasoning Wei et al. [2022b]. These methods guide large language models to think step-by-step about a given problem through instructions. During forming a reasoning chain, the model needs to plan the solution approach to the problem and answer the decomposed sub-questions, forming a sub-question-answer intertwined reasoning chain result. John Sweller, a cognitive psychologist at the University of New South Wales, proposed the Cognitive Load Theory in 1988 Sweller [1988]. He argued that human cognitive capacity in processing new information is quite limited due to the restricted capacity of working memory. Therefore, we need to consider how to present information efficiently to reduce the load on working memory. Considering the contextual learning ability and the limited contextual learning window of language models Dong et al. [2023], cognitive overload often occurs during the formation of reasoning chains, leading to hallucinations or ineffective reasoning processes. Thus, significantly reducing the cognitive load that large language models face during chain-of-thought reasoning to alleviate potential cognitive overload is a pressing issue.

This paper proposes the Self-Guide method, a self-planning-based reasoning enhancement method for large language models to enhance their reasoning abilities. This method helps large language models generate common sense knowledge and reasoning guidelines (Guidelines) for complex problems through prompts, allowing the model to rehearse its reasoning process via self-planning. By combining this with

Funded by: NSFC (No. 62106126, No. 62272093), Joint Project of Liaoning Provincial Department of Science and Technology (No. 2023-MSBA-081), and the Fundamental Research Funds for the Central Universities (No. N2416012).

the reasoning chain, it reduces the cognitive load exhibited by large language models during reasoning. Notably, we significantly improved the performance of language models on various reasoning tasks without fine-tuning the large language models or using external tools. Specifically, the contributions of this paper are as follows:

- We propose a new prompt-based learning method, where the large model itself simulates the problem-solving process to generate corresponding reasoning guidelines for the given problem.
- By generating these guidelines, we alleviate the cognitive load issue inherent in traditional chain-of-thought reasoning models when solving complex problems, improving the performance and reliability of language model reasoning.
- We demonstrate through experiments that the problem reasoning guidelines generated by larger models can effectively guide smaller models to produce more accurate results, compensating for their shortcomings in reasoning tasks.

Our experiments implemented the Self-Guide method using gpt-3.5-turbo-1106, and it significantly outperformed traditional chain-of-thought-based reasoning models on four common reasoning tasks—language understanding, multi-hop question answering, temporal reasoning, and relational reasoning (achieving an average improvement of about 2%). Furthermore, we demonstrated through experiments the generalizability of the Self-Guide method on models with weaker reasoning capabilities (such as LLaMA-7B-Chat and LLaMA-13B-Chat), achieving an improvement of over 10% under conditions of weak inherent reasoning abilities. All our data and code are open-sourced on GitHub.

2 Related Work

Complex reasoning provides opportunities for constructing numerous applications based on language models, potentially making language models the next-generation ”operating system” OpenAI et al. [2024]. Therefore, large language models should possess strong logical reasoning abilities for complex reasoning tasks and be capable of completing complex instructions through interaction with tools, users, and all elements of the external environment. Despite their excellent performance on many tasks, multi-step reasoning remains one of their weaknesses Valmeekam et al. [2022], Huang and Chang [2022]. To enhance the reasoning abilities of large language models, various methods have been proposed in prior research: Nye et al. proposed fine-tuning language models by generating “Scratchpads” (i.e., intermediate steps) before producing the final answer, enabling the models to synthesize/execute multi-step reasoning Nye et al. [2021]. Chain-of-Thought (CoT) prompts guide large language models to provide reasoning steps and intermediate results through instructions such as “Let’s think step by step” enhancing the final output’s accuracy. Although CoT models perform well with larger models, their effectiveness is less impressive with smaller models Wei et al. [2022b]. Additionally, inducing language models to generate reasoning explanations often requires constructing large datasets or using few-shot prompts that sacrifice accuracy. To address this, Zelikman et al. proposed Self-Taught Reasoner (STaR), leveraging the existing reasoning capabilities of large language models to iteratively generate high-quality reasoning explanations before providing answers Zelikman et al. [2022]. Wang and Huang et al. suggested that large language models have the ability to self-improve answers, proposing the Self-Consistency decoding strategy that uses CoT to generate multiple reasoning paths and answers, ultimately selecting the most frequent answer as the final output Wang et al. [2022a], Huang et al. [2022].

Although methods involving fine-tuning with high-quality data or few-shot prompts significantly enhance the complex reasoning abilities of large language models, they often face insufficient or low-quality data in practical scenarios, and these methods have low generalizability. To address this issue, Kojima et al. demonstrated that large language models still possess excellent reasoning abilities under Zero-Shot-CoT prompts Kojima et al. [2022]. However, CoT prompts often exhibit inconsistencies between reasoning and answers for difficult questions. To overcome the challenge of generalization from easy to difficult problems, Zhou et al. proposed the Least-to-Most prompting method, decomposing complex problems into a series of simpler sub-problems and solving them sequentially Zhou et al. [2022].

To fully utilize the vast parameter knowledge of large language models, Liu et al. suggested enhancing common-sense reasoning abilities by generating knowledge as additional input Liu et al. [2021].

When solving complex problems, the hallucination issue of language models is another pressing problem. Due to noisy or incorrect knowledge remembered during pre-training and the lack of prior knowledge in specific tasks, large language models may generate text that does not conform to facts or follow the original text, i.e., the hallucination problem of large language models Ji et al. [2023]. To mitigate this issue, researchers have proposed various methods: retrieval augmentation has been shown to significantly reduce hallucinations. Further, Peng et al. suggested that leveraging external knowledge and self-feedback from the language model can alleviate hallucinations Peng et al. [2023]. Many scholars believe that prompting large language models to provide step-by-step reasoning processes before giving answers can reduce hallucinations Wei et al. [2023], Yao et al. [2022], Wang et al. [2022b], Li et al. [2022], but this assumption only holds if the generated reasoning aligns with the model's actual reasoning process Jacovi and Goldberg [2020].

However, the reasoning generated by language models does not always remain consistent with their actual reasoning process Turpin et al. [2024]. To address this, Zhao et al. proposed the Verify-and-Edit model, a knowledge-enhanced CoT prompting framework that allows language models to edit the reasoning chain using external knowledge to improve the accuracy of predictions Zhao et al. [2023]. Lyu et al. proposed Faithful CoT, which converts the reasoning process represented in natural language into symbolic language for solving, ensuring higher fidelity in reasoning Lyu et al. [2023]. Although the above methods have achieved good results, traditional methods often require language models to simultaneously handle problem planning and answer generation, which may lead to cognitive overload during reasoning. To address this, this paper proposes the Self-Guide model, which decouples problem planning and answer generation to enhance the reasoning performance of large models on complex problems.

3 Self-Guide: Enhancing LLM Reasoning with Self-Generated Guidance

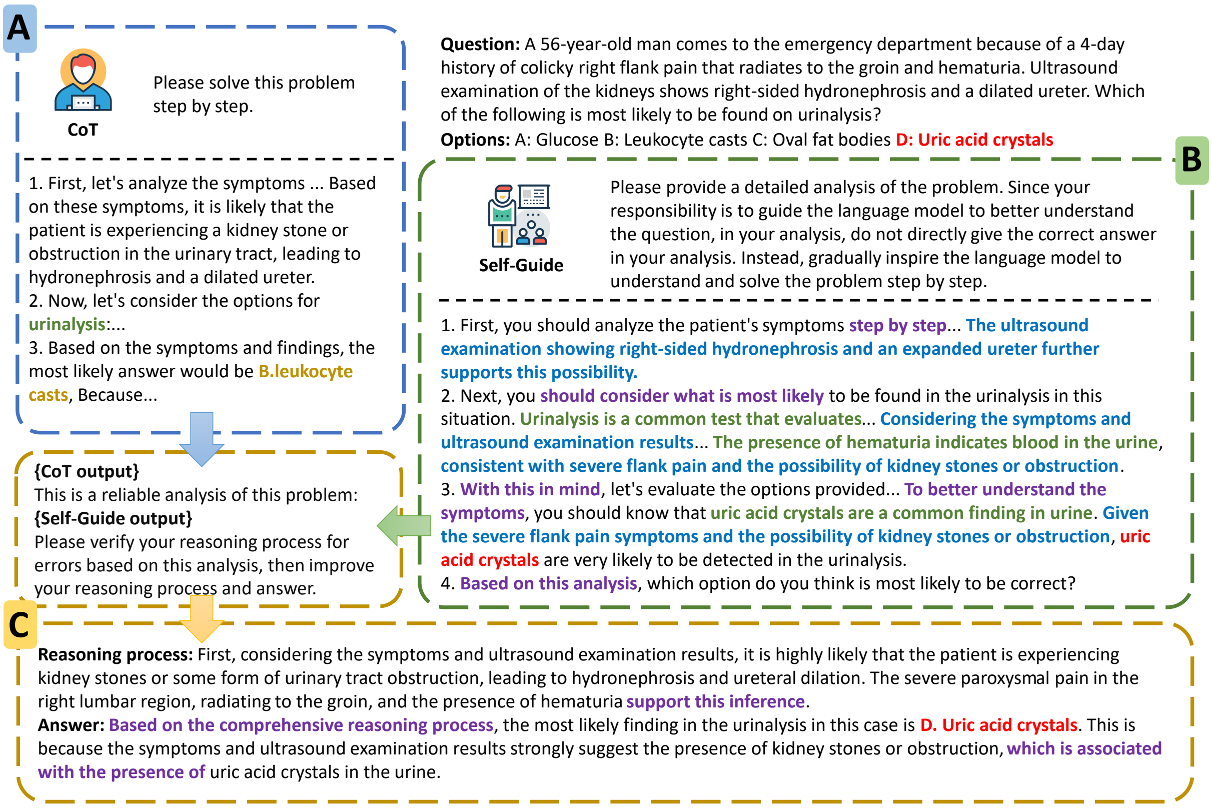
In this section, we introduce the Self-Guide method. Traditional Chain-of-Thought (CoT) enhanced reasoning methods maintain good generalization and reasoning capabilities but require simultaneous planning and answering during the reasoning process. The model needs to process and remember vast amounts of information throughout the reasoning process. Given the limitations of the reasoning time and the model's capabilities, large language models often face hallucination issues or ineffective reasoning. Effectively reducing cognitive load in the reasoning chain to alleviate potential cognitive overload during reasoning is a pressing issue.

Inspired by Cognitive Load Theory Sweller [1988], the Self-Guide model, as shown in Figure 1, aims to mitigate the cognitive load problem in traditional reasoning chain models. This method leverages the model's self-generated guidance to enhance the reasoning process by decoupling problem planning and answer generation. Overall, we first introduce the adaptive problem planning method based on large language models (3.1 section), then enhance the reasoning process using this adaptive planning (3.2 section).

3.1 Adaptive Problem Planning Method Based on Large Language Models

For a given input Q, the Self-Guide model enhances the reasoning performance of large models on complex problems through two steps: problem planning generation and enhancement based on problem planning. This method decouples problem planning and answer generation in traditional reasoning chain models to alleviate the cognitive load problem during complex reasoning, ultimately improving the reasoning performance of large models. The specific prompt templates are shown in Table 5.

Based on Cognitive Load Theory, we adopt a Zero-Shot setting to enable the large language model to adaptively plan the solution for a given problem. For a given problem Q, the model generates a planning Guideline. Specifically, it provides guidance based on the model's own experience to help the model reduce irrelevant cognitive load during reasoning, thus refocusing attention on the reasoning process. For the given problem Q, we generate the problem-solving plan using the large language model:



Please solve this problem step by step.

1. First, let's analyze the symptoms ... Based on these symptoms, it is likely that the patient is experiencing a kidney stone or obstruction in the urinary tract, leading to hydronephrosis and a dilated ureter.
2. Now, let's consider the options for urinalysis:...
3. Based on the symptoms and findings, the most likely answer would be B.leukocyte casts, Because...

Question: A 56-year-old man comes to the emergency department because of a 4-day history of colicky right flank pain that radiates to the groin and hematuria. Ultrasound examination of the kidneys shows right-sided hydronephrosis and a dilated ureter. Which of the following is most likely to be found on urinalysis?
Options: A: Glucose B: Leukocyte casts C: Oval fat bodies D: Uric acid crystals

Please provide a detailed analysis of the problem. Since your responsibility is to guide the language model to better understand the question, in your analysis, do not directly give the correct answer in your analysis. Instead, gradually inspire the language model to understand and solve the problem step by step.

1. First, you should analyze the patient's symptoms step by step... The ultrasound examination showing right-sided hydronephrosis and an expanded ureter further supports this possibility.
2. Next, you should consider what is most likely to be found in the urinalysis in this situation. Urinalysis is a common test that evaluates... Considering the symptoms and ultrasound examination results... The presence of hematuria indicates blood in the urine, consistent with severe flank pain and the possibility of kidney stones or obstruction.
3. With this in mind, let's evaluate the options provided... To better understand the symptoms, you should know that uric acid crystals are a common finding in urine. Given the severe flank pain symptoms and the possibility of kidney stones or obstruction, uric acid crystals are very likely to be detected in the urinalysis.
4. Based on this analysis, which option do you think is most likely to be correct?

{CoT output}
This is a reliable analysis of this problem:
{Self-Guide output}
Please verify your reasoning process for errors based on this analysis, then improve your reasoning process and answer.

Reasoning process: First, considering the symptoms and ultrasound examination results, it is highly likely that the patient is experiencing kidney stones or some form of urinary tract obstruction, leading to hydronephrosis and ureteral dilation. The severe paroxysmal pain in the right lumbar region, radiating to the groin, and the presence of hematuria support this inference.
Answer: Based on the comprehensive reasoning process, the most likely finding in the urinalysis in this case is D. Uric acid crystals. This is because the symptoms and ultrasound examination results strongly suggest the presence of kidney stones or obstruction, which is associated with the presence of uric acid crystals in the urine.

Figure 1: Overview of the Self-Guide model. The model consists of the Chain-of-Thought module (Step A), the adaptive problem-planning method based on large language models (Step B), and the reasoning enhancement method based on problem planning (Step C). The elements of background knowledge (Knowledge), problem understanding (Understanding), and problem planning (Planning) are highlighted accordingly.

$$\text{Guideline} = \text{Generate}(\text{Instruction\_Guide}, Q), \tag{1}$$

where Instruction_Guide represents the process of the model generating an adaptive problem planning Guideline based on the problem Q, and the specific prompt template is shown in Table 5. This method enables the model to generate problem planning adaptively, reducing cognitive load during complex reasoning and improving performance on complex reasoning tasks.

As shown in the output part of Step B in Figure 1, the generated Guideline includes background knowledge (Knowledge) as the basis for reasonable reasoning about the problem, providing relevant prior knowledge according to the specific domain of the problem. Having been trained on massive amounts of text, large language models possess extensive knowledge and can provide relevant background knowledge based on the problem content to help better understand and solve the problem. The problem understanding (Understanding) part requires the model to accurately grasp the implicit information and logical relationships of the problem, providing a foundation for subsequent reasoning and solving. The key is to extract effective information from the problem description and combine it with the ”background knowledge” to form a comprehensive understanding, thus targeting the reasoning and solving process. The problem planning (Planning) part is the core of the Self-Guide, where the model needs to develop a reasonable solution based on the knowledge and understanding of the problem. Through proper planning, the model can avoid blind reasoning and proceed methodically, improving problem-solving accuracy.

Parameter	checkpoint	temperature	top_p	max_seq_len	max_gen_len	max_batch_size
Setting Value	meta_llama	0.2	0.9	2048	128	8

Table 1: Parameter settings for LLaMA model during the inference process.

3.2 Reasoning Enhancement Method Based on Problem Planning

According to the experimental results of Wei and Lanham, smaller models often lack the ability to provide reasoning processes based on chain reasoning methods, such as LLaMA-7B and LLaMA-13B Wei et al. [2022b]. These models exhibit high intrinsic cognitive load during reasoning, and models start to demonstrate reasoning capabilities as the parameters increase to around 10B Lanham et al. [2023].

Given that the GPT-3.5 model has excellent basic reasoning capabilities, we adopted an adaptive learning process (see Figure 1). First, for the problem Q, we allow the model to independently use its cognitive resources to explore and solve the problem, generating an initial answer reasoning chain:

$$\text{CoT} = \text{Generate}(\text{Instruction\_CoT}, Q), \tag{2}$$

where the reasoning chain generation instruction Instruction_CoT is “Let’s think step by step.” Next, we enhance the model’s original reasoning chain using the Guideline generated by Equation 1. Specifically, we let the model review and reflect on the initial answer CoT based on the Guideline, correcting errors in the reasoning process and refining the reasoning process and final answer according to the Guideline:

$$\text{GuideCoT} = \text{Template}(\text{CoT}, \text{Guideline}, Q), \tag{3}$$

where the template is shown in the input part of Step C in Figure 1. Finally, the model generates a more dense and meticulous answer by combining independent thinking and self-planning learning processes. Through self-planning in Self-Guide, we reduce redundant reasoning steps in the model’s reasoning process, effectively lowering irrelevant cognitive load during reasoning. This method aims to maximize the model’s autonomy and reasoning capabilities, improving the learning outcomes and answer quality for complex reasoning problems by integrating self-planning and self-feedback.

4 Experiment Design

In this section, we describe the datasets, baselines, evaluation metrics, and implementation details used in our experiments.

4.1 Datasets

In our experiments, we evaluate the performance of the Self-Guide model on four tasks: knowledge understanding, multi-hop question answering, temporal reasoning, and relational reasoning. Due to the inference cost of ChatGPT, we randomly sampled 500 examples from the MMLU and CLUTRR datasets for testing, consistent with previous works Yoran et al. [2023], Zhang et al. [2024], Xu et al. [2024]. For the StrategyQA and Date Understanding datasets, which each contain fewer than 500 examples, we evaluated on the entire dataset.

Knowledge Understanding The knowledge understanding task aims to measure the knowledge acquired by the model during pre-training. We use the **MMLU** (Massive Multitask Language Understanding) dataset Hendrycks et al. [2021], which assesses the knowledge acquired during pre-training in zero-shot or few-shot scenarios. This benchmark covers 57 subjects, including science, technology, engineering, mathematics, humanities, and social sciences, ranging from elementary to advanced professional levels. It tests both world knowledge and problem-solving abilities, making it an ideal tool for identifying weak areas in model knowledge.

Multi-hop Question Answering We consider the **StrategyQA** dataset Geva et al. [2021]. This is an open-domain question dataset requiring implicit multi-step strategies to answer, where the reasoning steps are implicit within the question and should be inferred using strategies. For example, the question

”Did Aristotle use a laptop?” implicitly requires answering three sub-questions: “1. When did Aristotle live?”, “2. When was the laptop invented?”, and “3. Is the answer to sub-question 2 earlier than the answer to sub-question 1?”.

Temporal Reasoning We use the **Date Understanding** dataset from the **BIG-bench** benchmark bench authors [2023]. This dataset tests a language model’s ability to understand dates by asking questions about dates. It requires the model to infer dates based on relative periods. Given a context, the model should answer questions like “What is the date [day] in MM/DD/YYYY?”, where [day] could be today, tomorrow, etc. While this task is easy for humans, it is challenging for many language models, making it valuable for evaluating various optimization methods for reasoning.

Relational Reasoning We use the **CLUTRR** benchmark Sinha et al. [2019], a dataset for inferring kinship relations between characters in short stories. To successfully complete this task, models must extract relationships between entities and infer the logical rules governing these relationships. CLUTRR allows precise measurement of a model’s systematic generalization ability through the evaluation of logical rule retention combinations and robustness through the addition of curated noise facts.

4.2 Baselines

Under the Zero-Shot experimental setting, we compare the Self-Guide model with various prompt engineering methods. Our baselines mainly consist of standard zero-shot prompts and Chain-of-Thought (CoT) methods:

Standard Zero-Shot Prompting: Without providing examples, the model directly outputs the predicted answer for a given question through a specified instruction.

Chain-of-Thought (CoT): Building on the standard prompting method Brown et al. [2020], researchers require the model to explicitly output step-by-step intermediate reasoning steps before the final answer, enhancing the model’s common sense and reasoning abilities. CoT differs from traditional prompting by mapping <Input-Reasoning Chain-Output> instead of <Input-Output>. CoT includes Zero-Shot-CoT and Few-Shot-CoT. In our experiments, we use Zero-Shot-CoT as the baseline, adding ”Let’s think step by step” in the instruction without providing examples of intermediate reasoning steps Wei et al. [2022b].

4.3 Evaluation Metrics

Accuracy: We use accuracy (Acc) to evaluate model performance across different tasks, similar to previous work Xu et al. [2024]. We convert both model outputs and ground truth answers to lowercase and use string matching (StringEM) to calculate the accuracy between each model’s predicted result and the ground truth. Specific matching rules are detailed in Appendix B.

Perplexity: For evaluating the generalization capability of the Self-Guide method on different scales of language models, we use perplexity (PPL) to assess the quality of text generated during the reasoning process. For a tokenized sequence $X = (x_0, x_1, \ldots, x_t)$, the perplexity of sequence X is:

$$\mathrm{PPL}(X) = \exp\left\{ -\frac{1}{t} \sum_{i}^{t} \log p_\theta(x_i | x_{<i}) \right\}, \tag{4}$$

where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the i-th token given the preceding tokens $x_{<i}$ according to our model. Generally, lower PPL indicates higher textual coherence and fluency.

4.4 Implementation Details

For the Self-Guide method, as shown in Tables 2, 3, and 6, we use the OpenAI API to access the ChatGPT model for performance evaluation. Specifically, we use gpt-3.5-turbo-1106 as the base model for inference and set the temperature to 0.2 for generation. For evaluating the generalization of the Self-Guide method on LLaMA models, as shown in Table 4 and Figure 2, we first use gpt-3.5-turbo-1106 to generate the self-planning results (Guideline), then evaluate the performance of these results on the open-source LLaMA2-7B-Chat and LLaMA2-13B-Chat models. The parameter settings for the LLaMA models are detailed in Table 1.

Method	**MMLU**	**StrategyQA**	**Date**	**CLUTRR**
Zero-Shot	56.0	68.2	54.0	36.6
CoT	66.2	72.7	64.9	55.6
Self-Guide (No Instruct)	65.8	69.8	61.6	47.4
Self-Guide	67.4	70.2	61.3	48.8
CoT w. Self-Guide	**68.8**	**74.3**	**66.9**	**56.4**

Table 2: Experimental results of each model. Self-Guide is our proposed method. All methods use gpt-3.5-turbo-1106 as the base model for reasoning in this experiment. We present the experimental results of different models on various domains in the MMLU dataset in the appendix (Table 6).

5 Experiment

In this section, we present the overall performance of the Self-Guide model. Subsequently, we demonstrate the effectiveness of the Self-Guide method through ablation studies. We then delve into the generalization capability of the method on smaller models, the mechanisms of the Self-Guide method, and its characteristics. Finally, we conduct a case study.

5.1 Overall Performance of the Self-Guide Model

We present the comprehensive performance of the Self-Guide method on different domain datasets in Table 2. Under the Zero-Shot experimental setup, we compare the Self-Guide method with different types of baselines, including the standard Zero-Shot prompting method (Zero-Shot) and the Chain-of-Thought prompting method (CoT) under the Zero-Shot setting.

To investigate how large language models enhance reasoning through self-planning, we employed three methods to enhance model reasoning: direct self-planning reasoning (Self-Guide (No Instruct)), enhancing the reasoning process with self-planning on top of the standard prompting method (Self-Guide), and enhancing the reasoning process with self-planning on top of the Chain-of-Thought method (CoT w. Self-Guide).

As shown in Table 2, all three Self-Guide enhanced reasoning methods significantly outperform the baseline models (Zero-Shot) across all datasets. Notably, our model CoT w. Self-Guide demonstrates the best performance on all datasets. Compared to the Zero-Shot model, our model shows significant average accuracy improvements across all datasets (+6.1 to +19.8). Even compared to the CoT-enhanced reasoning model, our model shows certain degrees of improvement in average accuracy (+0.8 to +2.6), demonstrating the effectiveness of our model in addressing complex reasoning problems. This also shows that the Self-Guide method can alleviate cognitive load issues in traditional reasoning chains by generating natural language plans.

Given that the ChatGPT-3.5 model already possesses strong reasoning capabilities, the effectiveness of self-planning (Self-Guide and Self-Guide (No Instruct)) alone shows no significant improvement over the CoT model. This suggests that while the Self-Guide model can alleviate cognitive load issues by generating Guidelines, generating corresponding intermediate results for each step in the CoT model is also crucial.

5.2 Ablation Study

To further verify the effectiveness of the Self-Guide method, and whether the performance improvement brought by Self-Guide is due to enhanced reasoning capabilities through self-planning rather than simply the result of a second reasoning call, we conducted ablation experiments.

As shown in Table 3, we compare our model with different reasoning enhancement methods. We use the standard prompting method (Zero-Shot) and the Chain-of-Thought method (CoT) as base answers (Vanilla) and enhance the base answers with different methods, including self-reflection (Reflect) and checking the reasoning process of other language models (Debate).

As the experimental results show, compared to the baseline methods, allowing the model to reflect on its reasoning (Zero-Shot w. Reflect) slightly reduces reasoning ability (-21.1 to +0.6) compared to the

Method	Experimental Setup	MMLU	StrategyQA	Date	CLUTRR
Zero-Shot	Vanilla	56.0	68.2	54.0	36.6
	w. Reflect	56.6	47.1	51.0	24.0
	w. Debate	60.8	67.8	55.7	38.0
	w. Self-Guide	**67.3**	**70.2**	**61.3**	**48.8**
CoT	Vanilla	66.2	72.7	64.9	55.6
	w. Reflect	36.2	51.0	60.2	25.4
	w. Debate	63.2	65.3	61.3	36.4
	w. Self-Guide	**68.8**	**74.3**	**66.9**	**56.4**

Table 3: Results of ablation experiments. All methods use gpt-3.5-turbo-1106 as the base model for reasoning. Self-Reflect updates the answer through self-reflection after the language model generates the answer, and Self-Debate updates the answer after the language model checks the answers of other models.

Zero-Shot model. For the CoT model, the self-reflection (CoT w. Reflect) results in a more significant decrease in reasoning ability (-30.2 to -4.7). Allowing the model to check the reasoning process of other language models and provide its answer (Debate) slightly improves answer accuracy (-0.4 to +4.8) compared to the Zero-Shot model, but shows a performance decrease compared to the CoT model (CoT w. Debate) (-19.2 to -3.0).

The experimental results indicate that the Self-Guide method enhances the reasoning ability of large language models by alleviating cognitive load issues, rather than simply improving through self-reflection (Reflect) or self-checking (Debate). Compared to the Zero-Shot model, the Self-Guide method improves the reasoning ability across all datasets (+2.0 to +12.2), achieving performance comparable to the CoT model on some datasets. This demonstrates the critical role of self-planning results in handling logical reasoning problems.

Additionally, the Self-Guide method, when combined with CoT, further enhances model performance. The Self-Guide method improves reasoning ability across all datasets (+0.8 to +2.6). This indicates that the improvements achieved by the Self-Guide method are not merely due to a second reasoning call, but rather due to the model refining the reasoning chain through self-planning, enhancing the model's reasoning capabilities on complex tasks.

5.3 Generalization Capability of Self-Guide on Different Scale Models

This experiment demonstrates the generalization capability of the Self-Guide method on different models, as shown in Table 4. We use the Self-Guide method to enhance the reasoning abilities of smaller large language models, such as LLaMA2-7B-Chat and LLaMA2-13B-Chat, enabling these models to exhibit stronger reasoning capabilities. Since smaller LLaMA models have weaker basic reasoning abilities and cannot effectively generate reasoning chains or solve complex reasoning tasks, we consider ChatGPT-3.5 as a teacher model. Through the Self-Guide method, we use the CoT and Guideline generated by ChatGPT-3.5 to guide the smaller models in solving complex reasoning problems (GPT-Guide). Specifically, in the GPT-Guide method, we first use ChatGPT-3.5 to generate reasoning chains and Guidelines. After removing the final answers from these generated reasoning chains and Guidelines, we guide the LLaMA models using the reasoning chains (CoT) generated by ChatGPT-3.5, guide the models using the Guidelines (Guideline) generated by ChatGPT-3.5, and guide the LLaMA models using both the CoT and Guideline (CoT w. Guideline) generated by ChatGPT-3.5.

As shown in Table 4, the CoT w. Guideline method demonstrates significant performance across all tasks and datasets, especially in multi-hop question answering, temporal reasoning, and relational reasoning tasks. Specifically, compared to the method where the LLaMA model independently solves reasoning problems (LLaMA-Guide), on the LLaMA2-7B-Chat model, our method shows performance improvements of +5.9 to +26.5 over the standard Zero-Shot prompting method and +5.1 to +22.9 over the CoT prompting method. On the LLaMA2-13B-Chat model, the performance improvement ranges from

Method	**Experimental Setup**	**MMLU**	**StrategyQA**	**Date**	**CLUTRR**
LLaMA2-7B-Chat					
LLaMA-Guide	Zero-Shot	44.8	59.6	28.4	27.2
	CoT	40.8	60.4	32.0	34.6
GPT-Guide	CoT	52.0	64.7	52.9	37.6
	Guideline	48.8	**66.1**	44.3	45.0
	CoT w. Guideline	**53.0**	65.5	**54.9**	**45.6**
LLaMA2-13B-Chat					
LLaMA-Guide	Zero-Shot	46.8	62.2	39.0	34.4
	CoT	45.6	51.2	45.7	39.4
GPT-Guide	CoT	54.0	62.7	59.9	43.0
	Guideline	54.2	**65.9**	57.4	46.0
	CoT w. Guideline	**56.8**	64.9	**61.6**	**47.2**

Table 4: Experimental results of Self-Guide's generalization capability on different scale language models. The LLaMA-Guide method involves LLaMA models independently solving reasoning problems, while the GPT-Guide method involves gpt-3.5-turbo-1106 generating CoT and Guidelines, which are then used to guide LLaMA models to solve complex reasoning problems after removing the final answers.

+2.7 to +22.6 over the standard Zero-Shot prompting method and +7.8 to +15.9 over the CoT prompting method. These results indicate that using Guidelines generated by larger models based on the Self-Guide method to guide smaller models can effectively transfer the rich semantic representations and reasoning abilities of larger models to smaller models, compensating for their deficiencies in reasoning tasks.

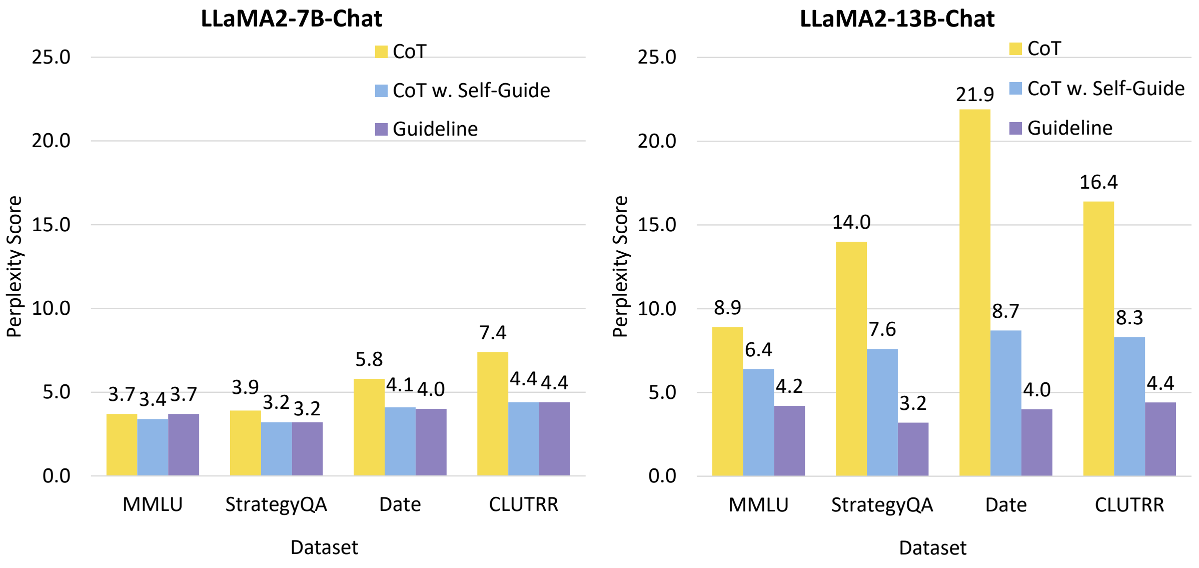


Figure 2: Differences in the perplexity of LLaMA models using different prompting methods.

Furthermore, in the method where the reasoning process of ChatGPT-3.5 guides LLaMA models in solving reasoning problems (GPT-Guide), the learning method guided by CoT w. Guideline shows performance improvements over the CoT-guided learning method. On the LLaMA2-7B-Chat model, the CoT w. Guideline guided model shows performance improvements ranging from +0.8 to +8.0 over the CoT guided model. On the LLaMA2-13B-Chat model, the CoT w. Guideline guided model shows performance improvements ranging from +1.7 to +4.2 over the CoT guided model. As model parameters decrease, the performance gap between the CoT and CoT w. Guideline methods widen, from 0.8% to

Question: A 56-year-old man comes to the emergency department because of a 4-day history of colicky right flank pain that radiates to the groin and hematuria. Ultrasound examination of the kidneys shows right-sided hydronephrosis and a dilated ureter. Which of the following is most likely to be found on urinalysis?

Options: A: Glucose B: Leukocyte casts C: Oval fat bodies D: Uric acid crystals

Step 1: Self-Planning Based on Self-Guide

Self-Guide Prompt

You are a knowledgeable and patient professor whose role is to guide students in solving problems correctly. Here is a question: {Question} Note: Since your responsibility is to guide students in answering the question, your analysis should think step by step, Please note that your role is to guide them step by step through the problem, so please don't give them the final result.

Generated Guideline

1. Alright, let's break this down step by step. First, let's consider the symptoms presented by the patient. ... The ultrasound examination showing ... supports this possibility.

2. Now, let's think about what would be most likely to be found on urinalysis in this case. Urinalysis is a common test that evaluates various components of urine, including the presence of blood, protein, glucose, and different types of cells and crystals. Given the symptoms and the ultrasound findings, ... The presence of hematuria suggests that there is blood in the urine, which is consistent with the colicky flank pain and the possibility of a kidney stone or obstruction.

3. Considering this, let's evaluate the options provided:... Therefore, we can eliminate option D. Uric acid crystals are a common finding in the urine of patients with kidney stones, especially those composed of uric acid. Given the symptoms of colicky flank pain and the possibility of a kidney stone or obstruction, uric acid crystals are the most likely finding on urinalysis in this case.

4. Based on this analysis, which option do you think is most likely correct?

Step 2: Final Answer Generation

Final Answer Prompt

{CoT-log}

This is another analysis of this question:{Self-Guide}. Please combine this reasoning process with your reasoning processes, then give me a better reasoning process.

Generation Format: 1. Inference process: 2. Answer:

Generated Final Answer

Inference process:

1. First, considering the symptoms and ultrasound findings, it is likely that the patient is experiencing a kidney stone or some form of obstruction in the urinary tract, leading to hydronephrosis and dilated ureter. The presence of colicky right flank pain, radiating to the groin, and hematuria supports this inference.

2. Next, let's evaluate the options provided for urinalysis: A...B... C... D...

Given the symptoms of colicky flank pain and the possibility of a kidney stone or obstruction, uric acid crystals are the most likely finding on urinalysis in this case.

Answer: Based on the combined reasoning processes, the most likely finding on urinalysis in this case is uric acid crystals (option D). This is because the symptoms and ultrasound findings strongly suggest the presence of a kidney stone or obstruction, which would be associated with the presence of uric acid crystals in the urine.

Table 5: Case study. We randomly selected an example from the MMLU dataset to demonstrate the effectiveness of the Self-Guide method, highlighting the answers and relevant contents. This is the original input and output in English, with the translated Chinese version presented in Table **??** in the main text.

2.6% on ChatGPT-3.5, to 1.7% to 4.2% on LLaMA2-13B-Chat, and to 0.8% to 8.0% on LLaMA2-7B-Chat. These results indicate that smaller models have less cognitive capacity, amplifying the cognitive load issues of planning and answering in CoT. The problem-solving approach in CoT is difficult to adapt to smaller models. The Guidelines generated by the Self-Guide model act more as a problem-solving strategy, guiding the model to consider necessary knowledge and steps, better fitting the capabilities of smaller models, and helping them generate better results.

To better quantify the intrinsic cognitive load during the reasoning process, as shown in Figure 2, we use perplexity (PPL) to measure the confidence of the language model in generating answers during the reasoning process. As the results indicate, models guided by Guidelines show significantly lower perplexity compared to other prompting methods. With LLaMA2-7B-Chat and LLaMA2-13B-Chat models, the PPL value consistently remains around 4.0 across all datasets (-0.8 to +0.4). This demonstrates that the Guidelines generated by the Self-Guide method can more easily help models of different scales generate correct answers, reducing model uncertainty. Furthermore, the Self-Guide method offers a potential way for larger models to guide smaller models' learning. Unlike distillation frameworks through reasoning chains Hsieh et al. [2023], distillation through generated Guidelines can better fit the learning process of smaller models, helping them understand the problem-solving approach, necessary knowledge, and planning of larger models, rather than blindly imitating the problem-solving approach of larger models.

5.4 Case Study

Finally, we randomly selected a case from the MMLU dataset to analyze the effectiveness of the Self-Guide method. Case studies from other datasets are detailed in Appendix D.

As shown in Table 5, the reasoning process of the model enhanced by the Self-Guide method is more concise and information-dense compared to the CoT model, with tighter logical relationships between reasoning steps. Specifically, in the chosen case, Self-Guide first identifies the problem domain (medical clinical diagnosis), combines the symptoms described in the problem with relevant prior knowledge (e.g., ultrasound examination, urine analysis), extracts key information (e.g., hematuria, right hydronephrosis), combines this with prior knowledge to form a deep understanding of the clinical condition, and finally provides an analysis plan starting from each option. After obtaining the Guideline through Self-Guide, the model can clearly outline the reasoning process for the entire problem, reflecting on and analyzing the initial reasoning chain by integrating the understanding provided by the Guideline, ultimately providing the correct answer. This demonstrates that Self-Guide fully utilizes the model's strong generative capabilities, allowing it to flexibly adjust its reasoning strategy based on the specific problem, thereby better handling various complex reasoning scenarios. Self-Guide helps the model reflect on the CoT results, guiding detailed problem analysis and forming knowledge-dense Guidelines, thereby reducing cognitive load during complex reasoning tasks.

6 Conclusion

This paper presents the Self-Guide method, which combines common sense knowledge and reasoning instructions generated by large language models, allowing the models to enhance their reasoning abilities through self-planning. This method significantly improves the reasoning capabilities of language models without fine-tuning them. We have demonstrated that our method outperforms baseline methods in four common reasoning tasks: language understanding, multi-hop question answering, temporal reasoning, and relational reasoning. The analysis of the experimental results further indicates that the Self-Guide model can effectively transfer the language representations and reasoning abilities of larger models to smaller models with weaker reasoning capabilities through the use of Guidelines. Additionally, it significantly reduces the uncertainty in the answers generated by the smaller models.

References

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

BIG bench authors. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. Transactions on Machine Learning Research, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=uyTL5Bvosj`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360, 2021.

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

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

Cheng-Yu Hsieh, Chun-Liang Li, CHIH-KUAN YEH, Hootan Nakhost, Yasuhisa Fujii, Alex Jason Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In The 61st Annual Meeting Of The Association For Computational Linguistics, 2023.

Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. Minicpm: Unveiling the potential of small language models with scalable training strategies, 2024.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. arXiv preprint arXiv:2210.11610, 2022.

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

Alon Jacovi and Yoav Goldberg. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? arXiv preprint arXiv:2004.03685, 2020.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. ACM Comput. Surv., 55(12), mar 2023. ISSN 0360-0300. doi: 10.1145/3571730. URL https://doi.org/10.1145/3571730.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. Advances in neural information processing systems, 35:22199–22213, 2022.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. arXiv preprint arXiv:2307.13702, 2023.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, Wenhu Chen, and Xifeng Yan. Explanations from large language models make small reasoners better, 2022.

Jiacheng Liu, Alisa Liu, Ximing Lu, Sean Welleck, Peter West, Ronan Le Bras, Yejin Choi, and Hannaneh Hajishirzi. Generated knowledge prompting for commonsense reasoning. arXiv preprint arXiv:2110.08387, 2021.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. Faithful chain-of-thought reasoning. arXiv preprint arXiv:2301.13379, 2023.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. arXiv preprint arXiv:2112.00114, 2021.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff

Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. Check your facts and try again: Improving large language models with external knowledge and automated feedback, 2023.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4506–4515, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1458. URL https://aclanthology.org/

D19-1458.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. arXiv preprint arXiv:2107.02137, 2021.

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

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. Advances in Neural Information Processing Systems, 36, 2024.

Karthik Valmeekam, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Large language models still can't plan (a benchmark for llms on planning and reasoning about change). arXiv preprint arXiv:2206.10498, 2022.

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

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, and Denny Zhou. Rationale-augmented ensembles in language models, 2022b.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. arXiv preprint arXiv:2206.07682, 2022a.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023.

Zhipeng Xu, Zhenghao Liu, Yibin Liu, Chenyan Xiong, Yukun Yan, Shuo Wang, Shi Yu, Zhiyuan Liu, and Ge Yu. Activerag: Revealing the treasures of knowledge via active learning, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. arXiv preprint arXiv:2210.03629, 2022.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. Answering questions by meta-reasoning over multiple chains of thought, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. Advances in Neural Information Processing Systems, 35:15476–15488, 2022.

Jintian Zhang, Xin Xu, Ningyu Zhang, Ruibo Liu, Bryan Hooi, and Shumin Deng. Exploring collaboration mechanisms for llm agents: A social psychology view, 2024.

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. arXiv preprint arXiv:2305.03268, 2023.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv:2205.10625, 2022.

Method	**Massive Multitask Language Understanding (MMLU)**				
	All	Hum.	Soc. Sci.	STEM	Other
Zero-Shot	56.0	47.2	67.8	46.3	68.3
CoT	66.2	59.0	74.8	63.4	71.3
Self-Guide	65.8	57.4	73.9	55.4	**82.7**
Zero-Shot w. Self-Guide	67.4	**62.6**	72.2	61.2	77.6
CoT w. Self-Guide	**68.8**	58.4	**76.5**	**65.9**	80.2

Table 6: Experimental results of different models on various domains in the MMLU dataset. Self-Guide is our proposed method. All methods use gpt-3.5-turbo-1106 as the base model for reasoning in this experiment.

Appendix

A Performance of Models on Different Domains in the MMLU Dataset

The MMLU benchmark covers 57 subjects, including science, technology, engineering, mathematics, humanities, and social sciences. Its difficulty ranges from elementary to advanced professional levels, testing both world knowledge and problem-solving abilities. The refinement and breadth of these subjects make the benchmark an ideal tool for identifying weak areas in model knowledge. We conducted experiments across various domains such as all subjects (All), humanities and ethics (Hum.), social sciences (Soc. Sci.), STEM, and other subjects (Other) (Table 6).

B Details of Evaluation Metrics

In calculating the accuracy between each model's predicted results and the standard answers using the StringEM method, we first convert both the model outputs and the standard answers to lowercase. For the MMLU and StrategyQA datasets, we extract the first word from the model's generated results as the predicted result for matching with the standard answer. For the Date dataset, we extract all patterns composed of digits, slashes, digits, slashes, and digits, convert them to the standard date format (MM/DD/YYYY), and match them with the standard answer. For the CLUTRR dataset, we directly perform string matching to calculate the accuracy between each model's predicted results and the standard answers.

C Case Study of the Original English Version

We randomly selected an example from the MMLU dataset to showcase the effectiveness of the Self-Guide method, highlighting the answers and relevant content. We present the original input and output of the case study in Table **??**. For ease of reading, we provide a summarized version of the case study in Chinese in the main text.

D Case Studies from Other Datasets

We randomly selected an example from both the CLUTRR and Date datasets to demonstrate the effectiveness of the Self-Guide method, highlighting the answers and relevant content. As the StrategyQA dataset requires submission to a leaderboard for evaluation, we cannot determine the correctness of individual examples and thus do not include a case study for StrategyQA.

Question: A 56-year-old man comes to the emergency department because of a 4-day history of colicky right flank pain that radiates to the groin and hematuria. Ultrasound examination of the kidneys shows right-sided hydronephrosis and a dilated ureter. Which of the following is most likely to be found on urinalysis?

Options: A: Glucose B: Leukocyte casts C: Oval fat bodies D: Uric acid crystals

Original Chain-of-Thought (CoT) Prompt

Here is a question:{Question} In order to solve this problem, analyze and reason step by step, and give your thinking and reasoning process while solving the problem.

Original Chain-of-Thought (CoT) Output

1. First, let's analyze the symptoms ... Based on these symptoms, it is likely that the patient is experiencing a kidney stone or obstruction in the urinary tract, leading to hydronephrosis and a dilated ureter.

2. Now, let's consider the options for urinalysis:

A...B... C... D...

3. Based on the symptoms and findings, the most likely answer would be B. leukocyte casts, Because...

Step 1: Self-Planning Based on Self-Guide

Self-Guide Prompt

You are a knowledgeable and patient professor whose role is to guide students in solving problems correctly. Here is a question: {Question} Note: Since your responsibility is to guide students in answering the question, your analysis should think step by step, Please note that your role is to guide them step by step through the problem, so please don't give them the final result.

Generated Guideline

1. Alright, let's break this down step by step. First, let's consider the symptoms presented by the patient. ... The ultrasound examination showing right-sided hydronephrosis and a dilated ureter further supports this possibility.

2. Now, let's think about what would be most likely to be found on urinalysis in this case. Urinalysis is a common test that evaluates various components of urine, including the presence of blood, protein, glucose, and different types of cells and crystals. Given the symptoms and the ultrasound findings, ... The presence of hematuria suggests that there is blood in the urine, which is consistent with the colicky flank pain and the possibility of a kidney stone or obstruction.

3. Considering this, let's evaluate the options provided:... Therefore, we can eliminate option D. Uric acid crystals are a common finding in the urine of patients with kidney stones, especially those composed of uric acid. Given the symptoms of colicky flank pain and the possibility of a kidney stone or obstruction, uric acid crystals are the most likely finding on urinalysis in this case.

4. Based on this analysis, which option do you think is most likely correct?

Step 2: Final Answer Generation

Final Answer Prompt

{CoT-log}

This is another analysis of this question:{Self-Guide}. Please combine this reasoning process with your reasoning processes, then give me a better reasoning process.

Generation Format: 1. Inference process: 2. Answer:

Generated Final Answer

Inference process:

1. First, considering the symptoms and ultrasound findings, it is likely that the patient is experiencing a kidney stone or some form of obstruction in the urinary tract, leading to hydronephrosis and dilated ureter. The presence of colicky right flank pain, radiating to the groin, and hematuria supports this inference.

2. Next, let's evaluate the options provided for urinalysis: A. Glucose in the urine is typically associated with diabetes and would not be expected in this case. B. Leukocyte casts - While indicative of inflammation or infection, it may not be the most likely finding in this case. C. Oval fat bodies - Not directly related to the symptoms and ultrasound findings presented by the patient. D. Uric acid crystals. Given the symptoms of colicky flank pain and the possibility of a kidney stone or obstruction, uric acid crystals are the most likely finding on urinalysis in this case.

Answer: Based on the combined reasoning processes, the most likely finding on urinalysis in this case is uric acid crystals (option D). This is because the symptoms and ultrasound findings strongly suggest the presence of a kidney stone or obstruction, which would be associated with the presence of uric acid crystals in the urine.

Table 7: Case study. We randomly selected an example from the MMLU dataset to demonstrate the effectiveness of the Self-Guide method, highlighting the answers and relevant contents. This is the original input and output in English, with the translated Chinese version presented in Table **??** in the main text.

Info: [Davis] played chess with his brother [Rick]. [Davis] was disappointed that his father, [Kent], wouldn't be at the play to see him perform. [Sharon] likes taking her son [Oliver] to baseball practice. She also likes taking her son [Rick] to soccer practice. [Dorothy] made cookies with her grandson [Davis]. [Davis]'s father [Kent] is not allowed to eat sugar.
Question: How is [Dorothy] related to [Oliver]?
Answer: grandmother

Original Chain-of-Thought (CoT) Prompt
To solve the problem, Please think and reason step by step, then answer. Question: {Question}
Generation Format: 1. Reasoning process: 2. Answer:

Original Chain-of-Thought (CoT) Output
Reasoning process:
1. [Davis] is the grandson of [Dorothy].
2. [Oliver] is the son of [Sharon].
3. There is no direct relationship mentioned between [Dorothy] and [Oliver].
Answer: There is no direct relationship between [Dorothy] and [Oliver] mentioned in the information.

Step 1: Self-Planning Based on Self-Guide

Self-Guide Promp
You are a knowledgeable and patient professor whose role is to guide students in solving problems correctly. Here is a question:{Question} please provide a detailed analysis. Note: Since your responsibility is to guide students in answering the question, your analysis should think step by step, Please note that your role is to guide them step by step through the problem, so please don't give them the final result.

Generated Guideline
Step 1: Identify the family members mentioned in the story:
- Davis - Rick - Kent - Sharon - Oliver - Dorothy
Step 2: Determine the relationships between the family members that are explicitly mentioned in the story: - Davis is the grandson of Dorothy - Rick is the brother of Davis - Kent is the father of Davis - Sharon is the mother of Oliver and Rick
Step 3: Identify any indirect relationships that may help us determine the relationship between Dorothy and Oliver: - Davis is the grandson of Dorothy - Kent is the father of Davis - Sharon is the mother of Oliver - Sharon is also mentioned as the mother of Rick
Step 4: Consider the relationships between the family members and how they might connect Dorothy and Oliver: - Since Davis is the grandson of Dorothy, and Kent is the father of Davis, it means that Kent is the son of Dorothy. This makes Dorothy the grandmother of Kent's children, including Oliver.
Step 5: Based on the relationships identified, we can infer that Dorothy is the grandmother of Oliver.
Now, based on the analysis, we can determine the relationship between Dorothy and Oliver.

Step 2: Final Answer Generation

Final Answer Prompt
{CoT-log}
This is a credible analysis of this question: {Self-Guide}. Please verify your reasoning process for errors based on this analysis, then refine your reasoning process and answer.
Generation Format: 1. Inference process: 2. Answer:

Generated Final Answer
Inference process:
1. Identify the family members mentioned in the story.
2. Determine the relationships between the family members that are explicitly mentioned in the story.
3. Identify any indirect relationships that may help determine the relationship between the two family members in question.
4. Consider the relationships between the family members and how they might connect the two family members in question.
5. Verify the reasoning process for errors based on the analysis and refine if necessary.
Answer: Based on the analysis, Dorothy is the grandmother of Oliver.

Table 8: Case study. We randomly selected an example from the Date dataset to demonstrate the effectiveness of the Self-Guide method, highlighting the answers and relevant contents. This is the original input and output in English.

Question: Jane was born on the last day of February in 2000. Today is her 16-year-old birthday. What is the date yesterday in MM/DD/YYYY?”
Answer02/28/2016

Original Chain-of-Thought (CoT) Prompt
To solve the problem, Please think and reason step by step, then answer. Question: {Question} Please format the date in the following manner: MM/DD/YYYY.
Generation Format: 1. Reasoning process: 2. Answer:

Original Chain-of-Thought (CoT) Output
Reasoning process:
1. Jane was born on February 29, 2000, which was a leap year.
2. Since today is her 16th birthday, we need to calculate the date 16 years ago.
3. 16 years ago from 2021 is 2005.
4. Since 2005 was not a leap year, February 29th did not exist. Therefore, her birthday would have been on February 28th.
Answer: 02/28/2005

Step 1: Self-Planning Based on Self-Guide

Self-Guide Prompt
You are a knowledgeable and patient professor whose role is to guide students in solving problems correctly. Here is a question:{Question} please provide a detailed analysis. Note: Since your responsibility is to guide students in answering the question, your analysis should think step by step, Please note that your role is to guide them step by step through the problem, so please don’t give them the final result.

Generated Guideline
Step1: Determine the current date. Today is Jane’s 16th birthday, so we need to calculate the current date. Since she was born on the last day of February in 2000, her birthday would be on February 29th, 2016 (since 2000 was a leap year).
Step2: Calculate yesterday’s date.To find yesterday’s date, we need to subtract 1 day from Jane’s birthday. Since February 2016 has 29 days, we need to subtract 1 from 29 to get yesterday’s date.
Step3: Consider the month and year after subtracting 1 from 29.
Step4: Finalize the answer now that we have calculated yesterday’s date. By following these steps, we can determine the date yesterday in MM/DD/YYYY format.

Step 2: Final Answer Generation

Final Answer Prompt
{CoT-log}
This is a credible analysis of this question: {Self-Guide}. Please verify your reasoning process for errors based on this analysis, then refine your reasoning process and answer.
Generation Format: 1. Inference process: 2. Answer: MM/DD/YYYY.

Generated Final Answer
Inference process:
1. The current date is February 29, 2016, as it is Jane’s 16th birthday and she was born on the last day of February in 2000, which was a leap year.
2. To find yesterday’s date, we subtract 1 day from February 29, 2016, which gives us February 28, 2016.
3. Considering the month and year, yesterday’s date is February 28, 2016.
Answer: 02/28/2016

Table 9: Case study. We randomly selected an example from the Date dataset to demonstrate the effectiveness of the Self-Guide method, highlighting the answers and relevant contents. This is the original input and output in English.