

Self-Hint Prompting Improves Zero-shot Reasoning in Large Language Models via Reflective Cycle

Jindou Chen Jidong Tian Yaohui Jin*
{goldenbean, frank92, jinyh}@sjtu.edu.cn
MoE Key Lab of Artificial Intelligence,
Shanghai Jiao Tong University, Shanghai, China

Abstract

Chain-of-Thought (CoT) has brought a fresh perspective to improve the reasoning ability of large language models (LLMs). To relieve the burden of manual design in CoT, Zero-shot CoT has pioneered a direct interaction with LLMs. Based on it, researchers attempt to optimize reasoning paths through various prompting approaches like reflection, selection, and planning. However, few studies have focused on the possibility of combining all these strategies through a cognitive theory. Inspired by experiential learning, this paper proposes a new zero-shot prompting method based on Kolb’s reflective cycle, named Self-Hint prompting. Specifically, Self-Hint prompting introduces an automated iterative interaction approach to simulate the conscious reflection process, which uses intermediate observations as hints to guide LLMs. We have conducted comprehensive experiments on various math reasoning benchmarks. The empirical results on GPT models demonstrate the effectiveness of our method. Proposed Self-Hint prompting consistently outperforms other zero-shot baselines.

Keywords: Large language models; Zero-shot reasoning; Reflective cycle; Conscious reflection

Introduction

Large language models (LLMs) have recently demonstrated extraordinary performance in various NLP tasks (Brown et al., 2020; Otter, Medina, & Kalita, 2020; Thoppilan et al., 2022; Chowdhery et al., 2022). However, their reasoning ability cannot be promoted through solely scaling up model size (Rae et al., 2021; Srivastava et al., 2022). In order to tackle this challenge, numerous methods have been proposed and shown remarkable effectiveness (Zhou et al., 2022; Kojima, Gu, Reid, Matsuo, & Iwasawa, 2022; Chen, Ma, Wang, & Cohen, 2022). Chain-of-Thought (CoT) is a groundbreaking approach that instructs LLMs to emulate “step by step” reasoning process like humans when solving reasoning tasks (Wei et al., 2022; Fu, Peng, Sabharwal, Clark, & Khot, 2022). However, CoT-based methods need numerous manual elaborate reasoning steps in prompts.

Furthermore, some zero-shot prompting methods, such as Zero-shot CoT (Kojima et al., 2022) and PS prompting (L. Wang et al., 2023), prompt LLMs to generate intermediate results for automatic execution of CoT. Despite the boosted reasoning performance, the generated intermediate reasoning steps could still be erroneous. Although several works attempt to ameliorate the problem, such as designing novel prompts or utilizing external tools, the lack of systematic integration of these methods prevents the formation of

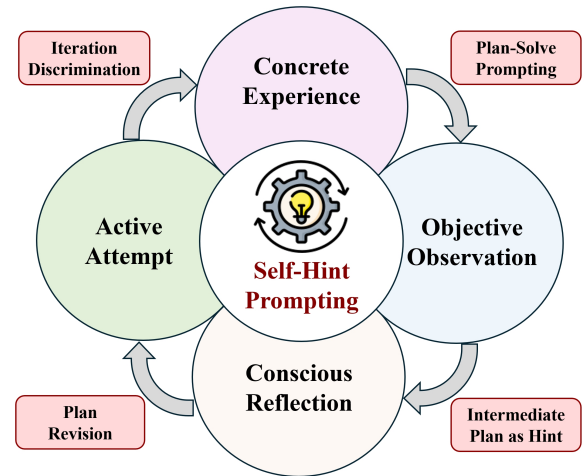


Figure 1: Framework of Self-Hint prompting

a unified cognitive framework. From a human perspective, when tackling a complex problem, people draw up a plan, continuously update the plan and iteratively analyze observations through conscious reflection. In experiential learning theory, Kolb’s reflective cycle exactly describes the spiral process of human immediate experience which results in reflections and observations of the experience. These reflections will lead to new actions to adjust to the experience that can be explored and tested (Kolb, 2014). Inspired by Kolb’s reflective cycle, we suggest that LLMs are able to execute the whole reflection cycle similar to humans.

To effectively stimulate the reflection ability of LLMs, we propose a simple but effective zero-shot prompting method, named **Self-Hint** prompting, shown in Figure 1. The specific process is: (1) LLMs pre-trained on massive amounts of data possess **concrete experience** for problem-solving. (2) After applying PS prompting (L. Wang et al., 2023), LLMs generate the initial intermediate plan and then solve the problem according to the plan, which will be conveyed back to LLMs and form the **objective observation**. (3) We combine the question and the devised plan as a hint to prompt LLMs to conduct **conscious reflection**, where we guide LLMs to consciously deal with the specific issues. (4) After revising the previous plan, LLMs take an **active attempt** to re-solve the problem according to reflective adjustments. We

*Corresponding author

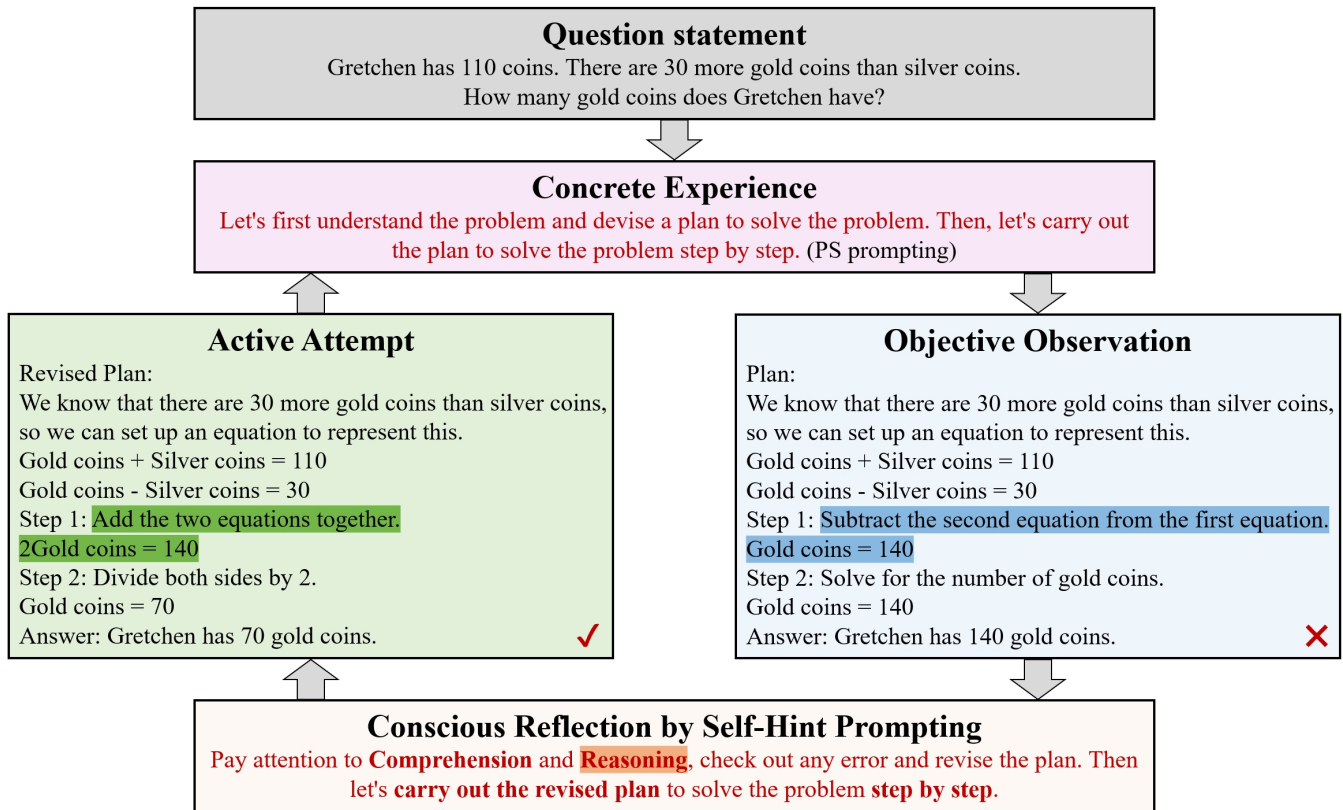


Figure 2: Example inputs and outputs of **Self-Hint** prompting and the problem is the 33rd question in the GSM8K testset.. Noticing that there is a reasoning issue shown in objective observation. LLMs take the wrong step to operate two equations, show in part. In conscious reflection, we explicitly guide LLMs to consciously fix reasoning issue, shown in part. In active attempt, LLMs succeed to rectify the issue after reflective cycle, shown in part.

repeat the whole process until the answer has convergence, which means the last two answers are consistent. In Self-Hint prompting, a critical challenge is how to induce LLMs to engage in conscious reflection. To identify the issues of initial execution, we conduct a detailed error analysis on 100 arithmetic test examples (Table 1). Results reveal two main pitfalls that should be considered in conscious reflections: comprehension issue and reasoning issue. The comprehension issue is defined as the failure in semantic understanding of the concept or problem context, and the reasoning issue is defined as the failure in generating the valid intermediate steps, such as missing steps or erroneous steps. Figure 2 exemplifies the proposed Self-Hint prompting on a reasoning issue.

We conduct experiments on four mathematical reasoning benchmarks, AQuA (Ling, Yogatama, Dyer, & Blunsom, 2017), SVAMP (Patel, Bhattamishra, & Goyal, 2021), GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). Compared to other zero-shot baselines, Self-Hint prompting achieves comparable or superior performance, even surpassing few-shot approaches on the AQuA dataset.

In summary, we make the following contributions:

- We propose a novel zero-shot method, Self-Hint prompting, to improve the LLMs' reasoning ability based on

Kolb's reflective cycle.

- In conscious reflection module, we propose a simple but effective prompting approach to tackle comprehension issue and reasoning issue.
- We demonstrate the effectiveness of Self-Hint prompting through extensive experimentation, including baseline comparisons and ablation study.

Related Works

Emergent Abilities and CoT based Prompting

An unpredictable phenomenon has been found regarding emergent abilities of LMs, which exclusively manifest in larger language models (Schaeffer, Miranda, & Koyejo, 2023). The capacity of LLMs to learn patterns from a few-shot input-output demonstrations within the given context enables it to successfully complete corresponding tasks, which is referred as in-context learning (Brown et al., 2020; Chowdhery et al., 2022; Shin, Razeghi, Logan IV, Wallace, & Singh, 2020). To exploit the multi-step reasoning ability in LLMs, CoT prompting (Wei et al., 2022) is a prominent work that appends multiple reasoning steps before outputting answer.

Subsequently, a series of works are dedicated to CoT improvement in different respects, such as prompt selection (Lu et al., 2022), prompt combination (X. Wang et al., 2022; Li et al., 2022; Fu et al., 2022; Weng, Zhu, He, Liu, & Zhao, 2022; “ChatAgri: Exploring potentials of ChatGPT on cross-linguistic agricultural text classification”, 2023), problem decomposition (Press et al., 2022; Khot et al., 2022; Zhou et al., 2022) and planning (Yao et al., 2023; Sun, Liu, Wang, Zhu, & Iyyer, 2023; Z. Wang, Cai, Liu, Ma, & Liang, 2023). Another research direction of promoting CoT is concentrated on rationales consistency. A novel research work is Self-Consistency (X. Wang et al., 2022), which samples various reasoning paths and chooses the most consistent one by majority vote. Progressive-Hint Prompting (C. Zheng, Liu, Xie, Li, & Li, 2023) (PHP) is also a typical one that uses the previously generated answers as hints and iteratively improve the quality of the model output in the few-shot setting. However, all these works require heavy prompt engineering.

Zero-shot CoT and Reflective Methods

To alleviate the burden of manual design in CoT, Zero-shot CoT (Kojima et al., 2022) is proposed to elicit reasoning step generation just by prompt “*Let’s think step by step*”. After analyzing the errors of Zero-shot CoT, Plan-and-Solve prompting (L. Wang et al., 2023) proposes a new zero-shot approach via asking LLMs to write intermediate plans to decompose a complex reasoning task into multiple reasoning steps. There are several works utilizing the reflective mechanism, where LLMs generate a critique for an answer, and the answer is iteratively refined. CRITIC (Gou et al., 2023) uses external tools to verify the whole generated answer. SELF-REFINE (Madaan et al., 2023) uses the same LLM as the generator, refiner and feedback provider to improve the result. A similar work to ours is RCI (Recursively Criticizes and Improves) (Kim, Baldi, & McAleer, 2023). Specifically, RCI consists of three stages: first, generate the initial answer by directly asking LLMs a question; second, criticize the generated answer (e.g. “*Review your previous answer and find problems with your answer*”); last, improve the answer based on the critique (e.g. “*Based on the problems you found, improve your answer*”). The different operations in our work are that we utilize the reflective cycle from experiential learning and stimulate LLMs’ reflection ability more effectively.

Self-Hint Prompting

Overview: The Kolb reflective cycle is a theoretical framework that describes how individuals learn based on the reflection of active experiences. As a problem-solving tool, it encourages individuals to reflect on their experiences, identify strengths and weaknesses, and develop strategies for improvement. Imaging when facing a complicated reasoning problem, people are not only able to draw up a plan to solve it by decomposing the problem into multiple steps for execution, but also reflect on experiences and revise the plan for improvement. Inspired by that, we introduce Self-Hint prompting, a new zero-shot prompting method to guide LLMs to

Method	PS	Self-Hint
Comprehension issue	23	20
Reasoning issue	14	12

Table 1: Error distribution of first 100 GSM8K problems by PS prompting and Self-Hint prompting with GPT model.

simulate humans’ reflective cycle. Figure 2 demonstrates the complete execution process using example inputs and outputs. LLMs succeed in improving problem-solving ability by sequentially utilizing previous outputs. Specifically, LLMs are first prompted to produce a base intermediate plan and initial answer based on their concrete experience, then combine the generated plan with the question as an objective observation for conscious reflection. After leading LLMs to rectify comprehension and reasoning issues, we prompt LLMs revise the plan for the next active attempt. We repeat the process when the last two answers are consistent, confidently indicating the correctness of the current answer. We will provide a detailed explanation on the four parts in the following.

Concrete experience

LLMs, pre-trained on extensive text datasets, acquire substantial knowledge and manifest emergent abilities to tackle multi-step reasoning problems with precision. Pre-trained datasets definitely include problem-solving text, which provides concrete experience for LLMs. To be more concrete, We follow PS prompting (L. Wang et al., 2023) and first put the input problem into a simple template “*Q: [I]. A: [PS]*”. Specifically, the input slot [I] contains the input problem text and [PS] is a zero-shot PS prompt (“*Let’s first understand the problem and devise a plan to solve the problem. Then, let’s carry out the plan and solve the problem step by step.*”). Therefore, knowledge of LLMs and the input prompt constitute **concrete experience** of our framework.

Objective observation

We input the above prompt to LLMs which subsequently output the intermediate plan and answer to the problem. To get an answer in the desired form, we extract the final answer from the output by appending the extra prompt (“*Therefore, the answer is*”). LLMs’ responses could be incorrect, analogous to errors that humans may make when addressing problems. According to Kolb’s reflective cycle, people can objectively observe past events for reflection. Hence, LLMs should review outputs for correction so that we keep the intermediate plans and the initial answer as **objective observation**.

Conscious reflection

Error analysis: In order to analyze the errors contained in objective observation, we conduct a detailed error analysis of generated intermediate plans on first the 100 GSM8K problems. Among all 37 incorrect cases, we follow (B. Wang et al., 2022; S. Zheng, Huang, & Chang, 2023) and ascribe errors of intermediate plans into two pivotal issues, shown

Datasets	Zero-shot				Few-shot	
	CoT	PS	RCI	Self-Hint	CoT	PHP
GSM8K	56.4	58.2	17.1	58.4 (+0.2)	58.4	57.5
AQuA	38.9	42.5	24.1	50.0 (+7.5)	48.4	44.4
SVAMP	69.9	72.0	64.2	72.6 (+0.6)	80.3	81.3

Table 2: Accuracy results of Self-Hint prompting and other baselines on three datasets. The results are from text-davinci-003 with greedy decoding. Baseline results are reported from their original papers.

Datasets	Zero-shot CoT	PS	RCI	Self-Hint	
GSM8K	70.1	76.6	59.7	77.1 (+0.5)	
AQuA	48.6	52.3	51.9	60.6 (+8.3)	
SVAMP	73.4	75.5	76.7	76.1 (-0.6)	
MATH	Algebra*	35.7	39.5	41.0	42.0 (+1.0)
	NumTheory*	18.5	19.8	19.0	25.7 (+5.9)
	Probability*	16.3	15.8	18.8	23.2 (+4.4)
	PreAlgebra*	45.0	43.5	46.8	47.5 (+0.7)
	InterAlgebra*	9.3	8.5	13.5	10.7 (-2.8)
	Precalculus*	10.3	10.5	11.0	10.3 (-0.7)

Table 3: Accuracy results of Self-Hint prompting and other baselines on four datasets. The results are from GPT-3.5-Turbo with greedy decoding. * means we randomly select 200 instances from the subset and run 3 restarts for stable results.

in Table 1: Comprehension issue and Reasoning issue. The comprehension issue is characterized by an inability to understand the concept or problem context semantically; and the reasoning issue is delineated as a failure in producing valid intermediate steps, encompassing missing and erroneous steps. For example, Figure 2 showcases the reasoning issue and LLMs generate the wrong operation for two equations in step 1 (e.g. “Subtract the second equation from the first equation.”). Just as humans can consciously engage in reflection, we should deliberately prompt LLMs to be attentive to these issues. Therefore, in order to simulate LLMs’ reflection ability and mitigate problems, our prompt template design of **conscious reflection** should satisfy two criteria:

(1) Prompt should elicit LLMs to detect comprehension and reasoning issues.

(2) Prompt should guide LLMs to revise the intermediate plan and implement it step by step.

For the first criterion, we add “*Pay attention to Comprehension and Reasoning*” as a trigger sentence to enable LLMs to understand the content of the problem as accurately as possible and mind errors resulting from wrong or missing reasoning steps. The utilization of such explicit prompt endows LLMs with the capacity for conscious reflection. For the second criterion, we include three important trigger instructions, “*check out any error and revise the plan*”, “*carry out the revised plan*” and “*step by step*”, to explicitly guide LLMs to correct the above issues, revise the previous intermediate plan and get into the next part, active attempt. Therefore, the structure of the whole Self-Hint prompt is “*Q: [I]. A: [IP]. Pay attention to Comprehension and Reasoning, check out errors*

and revise the plan. Then let’s carry out the revised plan to solve the problem step by step.”, where slot [IP] represents the intermediate plan on objective observation.

Active attempt

After prompting LLMs with conscious reflection, we are expected to get a modified intermediate plan and answer. Similar to the previous operation, we extract the answer using an extra prompt (“*Therefore, the answer is*”). Therefore, **active attempt** consists of the revised intermediate plan and answer.

To enhance the consistency and reliability, we propose an iterative framework. We will repeat the operations on the reflective cycle, using the revised intermediate plan in slot [IP] instead, until the stop condition is satisfied. There are two termination scenarios: the first is when two consecutive extracted answers are identical, signaling convergence of the answer. In this situation, we return it as the final answer. The second scenario is the number of interactions exceeds the pre-determined maximum number (e.g. $K = 5$ in our experiment, illustrated in Ablation Study). In such a case, we directly choose the last one as the final answer.

Experiments

Experimental Setups

Datasets: Following the prior studies, we evaluate Self-Hint prompting on four datasets, AQuA testset (Ling et al., 2017), SVAMP testset (Patel et al., 2021), GSM8K testset (Cobbe et al., 2021) and MATH testset (Hendrycks et al., 2021). For the first three, we evaluate whole testsets on text-davinci-003 and gpt-3.5-turbo model. For the MATH dataset, a more

No	Prompt Design	Datasets		
		AQuA	GSM8K*	SVAMP*
1	Check out any error and revise the plan. Then let’s carry out the revised plan to solve the problem step by step.	47.5	61.0	72.0
2	Pay attention to Comprehension and Reasoning , check out any error and revise the plan. Then let’s carry out the revised plan to solve the problem step by step.	50.0	63.0	73.5

Table 4: Performance comparison of different prompt selections tested on AQuA, GSM8K and SVAMP dataset with `text-davinci-003` model. * means we use the first 200 test instances of the dataset.

Iteration Number	datasets	
	AQuA	GSM8K*
$K = 0$	42.5	61.0
$K = 1$	46.8	60.5
$K = 3$	47.6	62.5
$K = 5$	50.0	63.0
$K = 7$	50.0	63.0

Table 5: Ablation study of iteration number K with `text-davinci-003` model. * means we select first 200 instances of the dataset.

challenging dataset, we only perform it on `gpt-3.5-turbo` model, randomly select 200 instances from each different subtopic (e.g. Algebra, Number Theory ...), run each subset 3 restarts for stable results and report the average accuracy.

Baselines: (1) Zero-shot baselines, including Zero-shot CoT (Kojima et al., 2022), PS prompting (L. Wang et al., 2023) and RCI (Recursively Criticizes and Improves) (Kim et al., 2023). RCI consists of three stages: first, generate the initial answer by directly querying LLMs a problem; second, ask LLMs to criticize the generated answer; last, prompt LLMs to improve the answer based on the critique. We reimplement RCI to follow our setting (e.g. maximum number of iterations is 5; use comparison between two consecutive answers to decide when to stop the loop). (2) Few-shot baselines: Manual-CoT (Wei et al., 2022) and Progressive-Hint Prompting (C. Zheng et al., 2023). The former designs eight hand-crafted examples as demonstrations and the latter adopts the previous answer to prompt LLMs. All methods adopt greedy decoding in our experiments.

Experimental Results

The main results of our study are presented in Table 2, 3. Table 2 reports the results on `text-davinci-003`. Self-Hint prompting outperforms other zero-shot baselines across all three datasets. We note that RCI performs poorly and consider two potential reasons: (1) the three-stage operational approach of RCI might be overly complex for `text-davinci-003`, especially the first step of direct question-answering methods probably introduces numerous

errors. The use of more powerful LLMs probably alleviates this issue. (2) the process of direct critique on the answer might include useless redundant information and lead to error amplification in the following steps. Therefore, our conscious reflection design is more practical than recursively criticizing of RCI. Despite the unfairness of this comparison, Self-Hint prompting achieves comparable results on the GSM8K dataset and better results on the AQuA dataset than few-shot methods. Therefore, we suggest that designing proper prompts to stimulate LLMs’ reflective ability might be a more effective pathway. This may spark the investigation of alternative approaches that minimize manual CoT design labor.

Table 3 presents the results of `gpt-3.5-turbo` model. Self-Hint prompting has achieved superior performance on the majority of the datasets, which means our method successfully stimulates the reflective capacity of LLMs. Although it performs slightly worse than RCI on InterAlgebra and Precalculus subset, we consider the possible reason is the inherent difficulty of the task, which leads to numerous errors in the generated intermediate plans so that LLMs are insufficient to correct these errors. It is worth noting that, compared to the previous experiments, RCI performs normally in this task. We speculate that though the questions are asked directly, a more powerful model (`gpt-3.5-turbo`) can generate valid critique according to the answer.

Ablation Study

We focus on the two aspects of Self-Hint prompting (reflection prompting and maximum number of iterations) and perform an ablation study. The former is to analyze the differences between conscious and unconscious reflection, while the latter reflects the efficiency of Self-Hint prompting.

Conscious reflection: To understand the design of conscious reflection, we perform an ablation study by removing the conscious prompting segment (“*Pay attention to Comprehension and Reasoning*”). Table 4 presents the results of the ablation experiment. Obviously, when we include the concrete trigger sentence, LLMs are able to achieve better performance across all three datasets. Therefore, this precise explicit instruction can indeed guide the attention of LLMs towards two vital issues, alleviating the corresponding errors. Therefore, the design of the conscious reflection module aiming at comprehension and reasoning issue is effective.

Problem	Josh decides to try flipping a house. He buys a house for \$80,000 and then puts in \$50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?		
Method	PS	Self-Hint (Unconscious reflection)	Self-Hint (Conscious reflection)
Prompt	Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan and solve the problem step by step.	Check out any error and revise the plan. Then let's carry out the revised plan to solve the problem step by step.	Pay attention to Comprehension and Reasoning, check out any error and revise the plan. Then let's carry out the revised plan to solve the problem step by step.
Output	<p>Plan:</p> <p>Step 1: Calculate the original value of the house. Original value = \$80,000</p> <p>Step 2: Calculate the increased value of the house after repair. Increased value = original value \times 150% = \$80,000 \times 1.5 = \$120,000</p> <p>Step 3: Calculate the total cost of the house and repairs. Total cost = Original value + Repairs = \$80,000 + \$50,000 = \$130,000</p> <p>Step 4: Calculate the profit. Profit = Increased value - Total cost = \$120,000 - \$130,000 = -\$10,000</p> <p>Therefore, Josh made a loss of \$10,000 on his house flipping venture.</p>	<p>Revised plan:</p> <p>Step 1: Calculate the increased value of the house: \$50,000 \times 150% = \$75,000</p> <p>Step 2: Calculate the new value of the house: \$80,000 (purchase price) + \$75,000 (increased value) = \$155,000</p> <p>Step 3: Calculate the total cost: \$80,000 (purchase price) + \$50,000 (repair cost) = \$130,000</p> <p>Step 4: Calculate the profit: \$155,000 (new value) - \$130,000 (total cost) = \$25,000</p> <p>Answer: Josh made a profit of \$25,000.</p>	<p>Revised Plan:</p> <p>Step 1: Calculate the original value of the house. Original value = \$80,000</p> <p>Step 2: Calculate the increased value of the house after repair. Increased value = original value + (original value \times 150%) = \$80,000 + (\$80,000 \times 1.5) = \$200,000</p> <p>Step 3: Calculate the total cost of the house and repairs. Total cost = Original value + Repairs = \$80,000 + \$50,000 = \$130,000</p> <p>Step 4: Calculate the profit. Profit = Increased value - Total cost = \$200,000 - \$130,000 = \$70,000</p> <p>Therefore, Josh made a profit of \$70,000 on his house flipping venture.</p>

Table 6: Case study of PS prompting, Self-Hint prompting with unconscious and conscious reflections, respectively.

Number of iterations: To understand the efficiency of Self-Hint prompting, we design an ablation study on number of iteration by setting upper limits to truncate the iterative process. We set different maximum iteration number K in $[0, 1, 3, 5, 7]$ and experiment on AQuA and GSM8K based on `text-davinci-003`. When $K = 0$, the reflection module is removed and the method degenerates into PS prompting. As shown in Table 5, larger iteration counts lead to higher accuracy performance. The phenomenon is consistent with intuition: the more interactions, the better the revised plan. It is worth noting that when $K = 1$, Self-Hint prompting does not necessarily outperform PS prompting because a single reflection lacks iteration discrimination, causing LLM to potentially alter the correct answer. In addition, when the value of K changes from 5 to 7, the result remains unchanged. This phenomenon illustrates that when the maximum number of iterations (K) reaches a finite small value, the iterative process converges completely. In this work, $K = 5$ almost ensures convergence of LLMs on all datasets. This convergence also indicates that the design of Self-Hint prompting does not significantly impact the efficiency.

Case Study

Continuing previous error analysis, we observe a valid alleviation of two issues after applying Self-Hint prompting in Table 1. The number of comprehension issue and reasoning issue has decreased by 3 and 2, respectively. To further investigate the specific reasons of why the conscious reflection module works, we choose a typical case with solutions of PS

Prompting, Self-Hint prompting with unconscious and conscious reflections, shown in Table 6. The case indicates that PS Prompting misunderstands the problem context (“This increased the value of the house by 150%”), which is a comprehension issue and leads to the loss of the original house value during calculation. Although Self-Hint prompting with unconscious reflection resolves the loss, it miscomprehends the increased value of the house as repairs value. However, the conscious instruction phrase “Pay attention to Comprehension and Reasoning” can indeed lead LLMs to compute “increased value” correctly. This result encourages that conscious reflection in LLMs is still worth further investigation and we leave it for future work.

Conclusion

Taking inspiration from Kolb’s reflective cycle, we suggest that LLMs can simulate the human reflection process to correct errors and introduce a simple but effective zero-shot method, Self-Hint prompting. Through identifying the two significant pitfalls, we provide LLMs with conscious reflection ability by explicit prompt. The effectiveness of Self-Hint prompting has been demonstrated through evaluations on various reasoning benchmarks. We believe that progressively optimizing the output of LLMs through the reflective cycle is an important step towards automatic iterative interaction with LLMs. How to foster LLMs self-reflection ability for specific tasks will be a significant direction of prompt engineering. We hope our works could motivate future research in this direction in the era of Large Language Models.

Acknowledge

This work was supported by Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

References

- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... others (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877–1901.
- Chatagri: Exploring potentials of chatgpt on cross-linguistic agricultural text classification. (2023). *Neurocomputing*, 557, 126708.
- Chen, W., Ma, X., Wang, X., & Cohen, W. W. (2022). Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*.
- Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... others (2022). Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., ... others (2021). Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Fu, Y., Peng, H., Sabharwal, A., Clark, P., & Khot, T. (2022). Complexity-based prompting for multi-step reasoning. *arXiv preprint arXiv:2210.00720*.
- Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Duan, N., & Chen, W. (2023). Critic: Large language models can self-correct with tool-interactive critiquing. *arXiv preprint arXiv:2305.11738*.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., ... Steinhardt, J. (2021). Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.
- Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., & Sabharwal, A. (2022). Decomposed prompting: A modular approach for solving complex tasks. *arXiv preprint arXiv:2210.02406*.
- Kim, G., Baldi, P., & McAleer, S. (2023). Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.
- Kolb, D. A. (2014). *Experiential learning: Experience as the source of learning and development*. FT press.
- Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., & Chen, W. (2022). On the advance of making language models better reasoners. *arXiv preprint arXiv:2206.02336*.
- Ling, W., Yogatama, D., Dyer, C., & Blunsom, P. (2017). Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.
- Lu, P., Qiu, L., Chang, K.-W., Wu, Y. N., Zhu, S.-C., Rajpurohit, T., ... Kalyan, A. (2022). Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *arXiv preprint arXiv:2209.14610*.
- Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegrefe, S., ... others (2023). Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.
- Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. *IEEE transactions on neural networks and learning systems*, 32(2), 604–624.
- Patel, A., Bhattamishra, S., & Goyal, N. (2021). Are nlp models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*.
- Press, O., Zhang, M., Min, S., Schmidt, L., Smith, N. A., & Lewis, M. (2022). Measuring and narrowing the compositionality gap in language models. *arXiv preprint arXiv:2210.03350*.
- Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, F., ... others (2021). Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.
- Schaeffer, R., Miranda, B., & Koyejo, S. (2023). Are emergent abilities of large language models a mirage? *arXiv preprint arXiv:2304.15004*.
- Shin, T., Razeghi, Y., Logan IV, R. L., Wallace, E., & Singh, S. (2020). Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.
- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A. A. M., Abid, A., Fisch, A., ... others (2022). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.
- Sun, S., Liu, Y., Wang, S., Zhu, C., & Iyyer, M. (2023). Pearl: Prompting large language models to plan and execute actions over long documents. *arXiv preprint arXiv:2305.14564*.
- Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., ... others (2022). Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.
- Wang, B., Min, S., Deng, X., Shen, J., Wu, Y., Zettlemoyer, L., & Sun, H. (2022). Towards understanding chain-of-thought prompting: An empirical study of what matters. *arXiv preprint arXiv:2212.10001*.
- Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., & Lim, E.-P. (2023). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091*.
- Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Narang, S., ... Zhou, D. (2022). Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.
- Wang, Z., Cai, S., Liu, A., Ma, X., & Liang, Y. (2023). Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task

- agents. *arXiv preprint arXiv:2302.01560*.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Chi, E., Le, Q., & Zhou, D. (2022). Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.
- Weng, Y., Zhu, M., He, S., Liu, K., & Zhao, J. (2022). Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.
- Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*.
- Zheng, C., Liu, Z., Xie, E., Li, Z., & Li, Y. (2023). Progressive-hint prompting improves reasoning in large language models. *arXiv preprint arXiv:2304.09797*.
- Zheng, S., Huang, J., & Chang, K. C.-C. (2023). Why does chatgpt fall short in providing truthful answers?.
- Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... Chi, E. (2022). Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.