Mirror: A Multiple-perspective Self-Reflection Method for Knowledge-rich Reasoning

Anonymous ACL submission

Abstract

While Large language models (LLMs) have the capability to iteratively reflect on their own outputs, recent studies have observed their struggles with knowledge-rich problems without access to external resources. In addition to the inefficiency of LLMs in self-assessment, we also observe that LLMs struggle to revisit their predictions despite receiving explicit negative feedback. Therefore, We propose Mirror, a **M**ulti**i**ple-per**r**spective self-**r**eflection method for kn**o**wledge-rich **r**easoning, to avoid getting stuck at a particular reflection iteration. Mirror enables LLMs to reflect from multiple-perspective clues, achieved through a heuristic interaction between a Navigator and a Reasoner. It guides agents toward diverse yet plausibly reliable reasoning trajectory without access to ground truth by encouraging (1) diversity of directions generated by Navigator and (2) agreement among strategically induced perturbations in responses generated by the Reasoner. The experiments on five reasoning datasets demonstrate that Mirror's superiority over several contemporary self-reflection approaches. Additionally, the ablation study studies clearly indicate that our strategies alleviate the aforementioned challenges.

1 Introduction

Large Language Models (LLMs) have become an important and flexible building block in a variety of tasks. They can be further improved by iterative correction in many tasks (Madaan et al., 2023; Gou et al., 2023a; Shinn et al., 2023; Pan et al., 2023), such as code generation, arithmetic problem solving and reasoning. During iterative refinement, the critic module, which assesses the current response and generates valuable feedback, is crucial to drive performance improvement.

Some research shows that LLMs have self-assessment abilities (Manakul et al., 2023; Madaan et al., 2023). For example, LLMs can reject its own prediction and generate a response *'I don't know'* when they are not confident about their predictions (Kadavath et al., 2022). Empirical observations demonstrate LLMs' competence in various reasoning tasks, leading to the utilization of advanced LLMs to evaluate the predictions made by other models (Hao et al., 2023; Zhou et al., 2023; Liu et al., 2023b). However, recent studies suggest that relying directly on LLMs' judgements is not trustworthy and can lead to failures in knowledge-rich iterative reasoning (Huang et al., 2023). To guide LLMs through a reasoning loop, existing solutions either incorporate external resources to verify LLMs' outputs (Peng et al., 2023; Yao et al., 2023b), or train a critic module on labelled assessment datasets (Gou et al., 2023a; Zelikman et al., 2022). Furthermore, self-consistency is considered a robust unsupervised method to identify confident and reliable LLM outputs.

In self-refinement, the quality of generated feedback also plays a pivotal role. The Self-Refine method (Madaan et al., 2023) introduced task-specific metrics for multifaceted feedback generation, requiring LLMs to evaluate their outputs across various aspects, such as *fluency, engagement*, and *relevance* for the dialogue generation task. This process often heavily relies on human expertise, and generating effective feedback for reasoning tasks can be even more difficult as it is obscure to define the essential attributes for different problems. Providing overly general feedback fails to guide LLMs toward generating better outputs in subsequent iterations.

The inefficiency of self-assessment and feedback generation capabilities largely hinders the performance of iterative refinements. On one hand, as depicted in Figure 1, it is evident that in the absence of a ground truth reference, LLMs fail to consistently improve their predictions, indicating

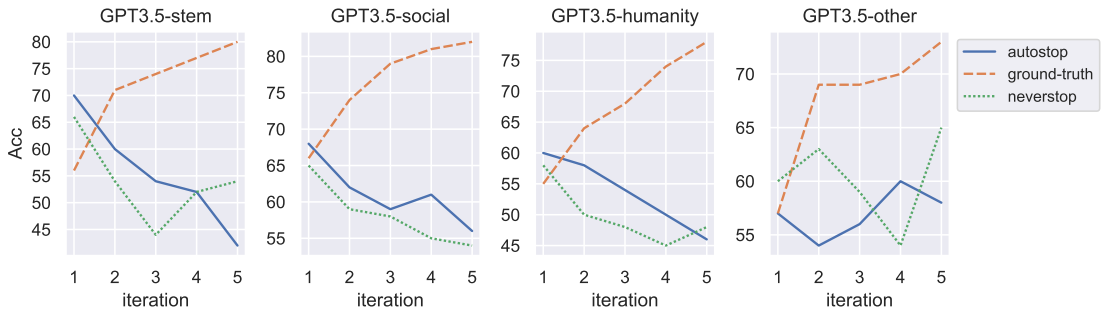


Figure 1: Without ground truth for validating LLM-generated outputs, LLMs struggle to consistently improve their own outputs due to their incapability of self-assessment. *Autostop* and *Neverstop* provide different generic feedback without leaking the correctness of the current response.

their limitations in self-assessment[1]. On the other hand, even when ground truth labels are available, LLMs often fail to adhere to instructions for revising their incorrect predictions, as shown in Figure 2. Each bar represents the number (averaged over 5 iterations) of revised (blue) and unchanged samples (grey) among the incorrectly predicted samples. It is undesirable to see that a large number of incorrect predictions stay unchanged, suggesting that LLMs can become trapped in a reasoning loop.

To address the aforementioned limitations and generate high-quality feedback without relying on human experts, we propose a novel framework, refer to as `Mirror` (**M**ulti**i**ple-pe**r**spective self-**r**eflection method for kn**o**wledge-rich **r**easoning). `Mirror` enables LLMs to reflect from multiple-perspective clues and this is achieved in a heuristic manner between a Navigator and a Reasoner, resembling a typical human tutoring process. For example, when tackling a complex scientific problem, the Navigator generates clues of key elements and rationales behind posing the question, which are crucial in focusing the response on the essential aspects. This information, tailored to the question, serve as instructions for prompting the Reasoner to adjust their predictions accordingly and avoid getting stuck at a particular stage.

To initiate the unsupervised self-reflection properly and avoid being trapped in the reasoning loop, `Mirror` integrates an intrinsically motivated planning algorithm to search for the optimal reasoning trajectory. Inspired by the findings in §3.1 and §3.2, we propose to reward both the diversity of generated directions and the agreement among strategically induced perturbations in responses. Notably differing from existing tree-based planning methods for reasoning (Hao et al., 2023; Zhou et al., 2023), `Mirror` avoids deteriorated searching space by encouraging diverse generative outcomes from LLMs at each reflection step, and enhances the self-assessment ability by considering the agreements among multiple-perspective perturbations strategically induced in responses. We evaluate the performance of `Mirror` on two categories of reasoning tasks: MMLU (Hendrycks et al., 2021), a knowledge-rich question-answering dataset, and FEVER (Thorne et al., 2018), a fact-checking dataset. `Mirror` achieves a significant average improvement of over 15% compared to recent popular unsupervised self-refinement methods. The empirical observations demonstrate that the proposed diversity-based reward and answer assessment strategy serve as reliable sources for performance enhancement.

2 Related Work

Self-Reflection LLMs. Extensive research (Honovich et al., 2022; Xie et al., 2023) has been conducted to enhance LLMs through the concept of self-reflection, where LLMs learn from automatically generated feedback to understand and reflect on their own outputs. This feedback can stem from various sources: the LLM itself (Madaan et al., 2023; Shinn et al., 2023), a separately trained critic module (Gou et al., 2023b; Peng et al., 2023) or external sources (Yao et al., 2023b), such as Wikipedia or an Internet Browser. Gou et al. (2023b); Peng et al. (2023) argued that evaluators trained on task-oriented feedback offer superior performance. For example, Refiner (Paul et al., 2023) took context and hypotheses as input to generate templates-based feedback for various error types. Recent studies (Peng et al., 2023; Shinn et al., 2023; Hao et al., 2023) have fully utilized the in-context learning capability of LLMs, prompting them to generate high-quality feedback based on their previous generation or potential templates. Madaan

[1]Details of *Autostop* and *Neverstop* are in Appendix A.1.

et al. (2023) proposed multiple task-oriented metrics and prompted LLMs to evaluate their own outputs based on these criteria. Similarly, Peng et al. (2023); Glaese et al. (2022) adopted external tools to predict multi-facet human preference scores. Our solution aligns with this trend by aiming to provide informative and customized instructions tailored to the specific task and query. Moreover, it seeks to achieve this without relying on human intervention or external tools, thereby rendering self-refinement more feasible in practice.

Reasoning models augmented with tree search. Recently, tree-based reasoning has attracted significant attention, such as Tree-of-Thought (ToT) (Yao et al., 2023a), Grace (Khalifa et al., 2023), and SelfEval-Decoding (Xie et al., 2023). At each reasoning step, ToT adopts breadth-first search and depth-first search, while the latter two methods select the top-k scoring candidates during the decoding process. Moreover, Monte-Carlo Tree Search (MCTS) is one of the popular search algorithms (Swiechowski et al., 2023), which strikes a balance between exploitation and exploration. Some existing approaches establish a reinforcement learning framework to maximize reward through learning optimal actions/states (Du et al., 2023a; Parthasarathy et al., 2023; Zhu et al., 2023). Other studies fully utilize the capability of LLMs for interaction and feedback generation. For instance, RAP (Hao et al., 2023) leveraged step-wise rewards from interactions with the world model to decompose and solve the problem step-by -step, rather than a iterative manner. LATS (Zhou et al., 2023) was the first work in leveraging MCTS for self-reflection. However, their feedback contains information from comparisons with ground truth, which is not applicable in our case. Instead, our approach, `Mirror` has no access to gold labels, and we incorporate a novel diversity reward to avoid the inefficient search in the reflection iteration.

3 Lost in the Reasoning Loop

Given the observed challenges in enhancing LLMs' self-improvement without ground truth labels, particularly in knowledge-rich reasoning tasks, our initial experiment aims to address these challenges by breaking them down into two sub-questions.

$\mathcal{Q}1$: *To what extent can LLMs assess the correctness of a statement?* This investigation involves enhancing their capabilities through supervised training. The primary goal is to discern if there are viable solutions to enhance the verification ability of LLMs on knowledge-rich statements.

$\mathcal{Q}2$: *How well can LLMs generate high-quality feedback to guide their own subsequent response update?* It is especially challenging when the feedback generation models are not trained on high-quality data, relying solely on the in-context learning capability of LLMs.

3.1 LLMs in Knowledge Grounding

We experiment with the multiple-choice dataset, MMLU (Hendrycks et al., 2021), covering 57 subjects across STEM, Humanity, Social and other domains. To evaluate the ability of LLMs in assessing the knowledge-rich statements, we construct the positive and negative statements by substituting the question with the correct choice and a randomly selected choice from the other three incorrect choices, respectively. Table 1 presents the assessment accuracy of assessing. There are three categories of methods: in-context learning, fine-tuned on statements, and classification based on intermediate activations from LLMs.

As illustrated in the first group results in Table A1, an increase in accuracy is observed as the size of Llama-2-13B-chat increases. Notably, GPT-3.5 with 175B parameters consistently achieves the best results across the three domains, although the improvement is not directly proportional to the parameter size. We then apply advanced prompting techniques, i.e., UniLangCheck (Zhang et al., 2023) on the best-performing method, GPT-3.5. Our analysis reveals that the improvements are predominantly driven by self-consistency, while UniLangCheck does not consistently contribute to improvement in grounding. For UniLangCheck, we firstly prompt LLMs to generate a fact about the key elements in a question before making the final assessment. It can be partially explained by the accumulation error, i.e., the inaccurate facts generated by LLMs before reaching the final conclusion can affect the outcome. We also calculate the correlation between accuracy and self-consistency, represented by the probability of generating a single answer through multiple repeated prompting. The average correlation R^2 for questions in the MMLU datasets across three LLMs is about 0.85, indicating that self-consistency can be relied upon as a proxy for assessment [2].

[2]Experiment details are shown in Appendix A.2, self-consistency evaluation results are shown in Table A1.

Model	STEM	Social	Humanity
Llama-2-13B-chat	0.541	0.540	0.525
Llama2-70B-chat	0.569	0.593	0.587
Vicuna-v1.5-13B	0.539	0.580	0.558
GPT-3.5(175B)	0.666	0.725	0.733
:+UniLangCheck	0.621	0.729	0.713
:+Self-Consistency	0.712	0.730	0.752
TRUE*	0.545	0.532	0.559
ActivationRegress*	0.531	0.529	0.553
ContrastSearch	0.606	0.645	0.617

Table 1: The (binary classification) accuracy in evaluating the factual correctness of statements in the MNLU dataset. Methods denoted with $\star$ can access to fact labels.

We also evaluate the performance of some *supervised methods* (denoted with $\star$ in Table 1). TRUE (Honovich et al., 2022) involves fine-tuning a T5 (Raffel et al., 2020) model on a collection of natural language inference (NLI) datasets for fact-checking. We further fine-tune its classifier head on our training set. ActivationRegress (Marks and Tegmark, 2023) trains classifiers using activations extracted from Llama2-13B 12-layer encodings as inputs. ContrastSearch (Burns et al., 2023) is trained using contrastive and consistency loss while having no access to the factual labels. This is achieved by constructing data pairs that include both a positive-labeled and negative-labeled statements, irrespective of the true factual labels. It is surprising that both TRUE and ActivationRegress are inferior than the unsupervised ContrastSearch.

3.2 LLMs in Feedback Generation

Evaluating the quality of generated feedback poses a significant challenge, particularly when such feedback is utilized across diverse tasks (Madaan et al., 2023). Drawing inspiration from the pivotal role of feedback in the self-improvement, we propose to leverage the performance of LLMs in subsequent iterations for evaluation. Specifically, LLMs can access to ground truth, enabling them to evaluate the correctness of their current responses. This information is then integrated into feedback generation. Consequently, we assess the quality of feedback by examining the percentage of examples that are incorrectly answered, along with the percentage of instances where responses in the next round are revised for the same incorrectly answered examples. This comparison sheds light on the effectiveness of instructions in guiding LLMs to rectify their erroneous responses. Firstly, we follow the settings in (Shinn et al., 2023) to incorporate the assessment results in the feedback: `"Observation: The answer is incorrect."` is inserted after presenting the question and previous attempt, and the LLMs are required to generate refection and response to this question again. From the results in Figure 2, it is consistently observed across different model scales that LLMs struggle to update their predictions despite receiving explicit negative feedback. The average percentage of successfully updated examples for GPT-3.5, Llama, and Vicuna are 65.6%, 51.79% and 74.09%, respectively, indicating an ample room for improvement.

Motivated by the following two observations: (1) LLMs are particularly susceptible to context influence at the beginning or near the end (Liu et al., 2023a), (2) In-Context Learning is highly sensitive to stylistic and emotional words in demonstrations (Min et al., 2022; Li et al., 2023), we develop three prompting strategies for feedback generation. An incorrectly predicted example with different prompting strategies is shown in Figure A2. The results in Table A2 and Table A3 suggest that based on correct question assessment, enhancing the exploration capability within a diverse answer space could lead to higher accuracy in answering knowledge-rich questions.

The above empirical findings regarding the two research questions provide valuable insights for our proposed model, named `Mirror`. Distinguishing itself from existing self-improvement methods, `Mirror` makes two significant contributions: (1) it features a Navigator module for generating multiple question-adaptive directions, with diversity constraints implemented to prevent invalid reflections. (2) it relies on the consistency of the inherent multiple perspectives for boosted self-assessment.

4 The Framework of Mirror

In this section, we introduce our unsupervised self-reflection framework, `Mirror`, depicted in Figure 3. The reward $\mathcal{R}$ consists of Diversity and Consistency terms. Diversity is applied to prevent reflection from becoming stuck and to facilitate intra-consistency involved in the stop criteria for self-assessment. The Consistency reward also influences direction generation.

4.1 Problem Setup

Given a question, the Reasoner is to arrive at the final answer through interacting with a Navigator. We consider a Markov Decision Process (MDP) defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \pi, \gamma, \mathcal{R})$, where the $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ denote the state and action, re-

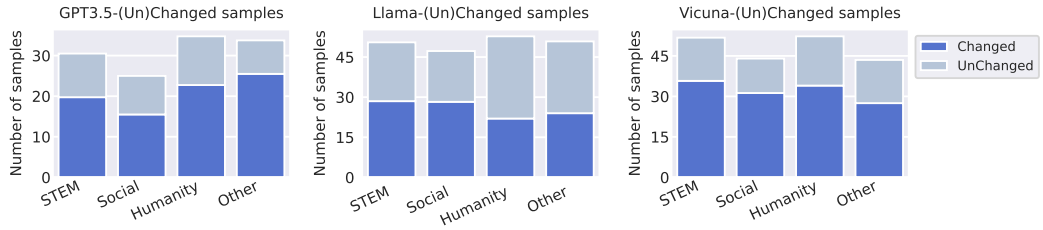


Figure 2: The average number (across all iterations) of changed and unchanged samples among those predicted incorrectly. Large percentage of unchanged samples indicate the limited capability for efficient reflection.

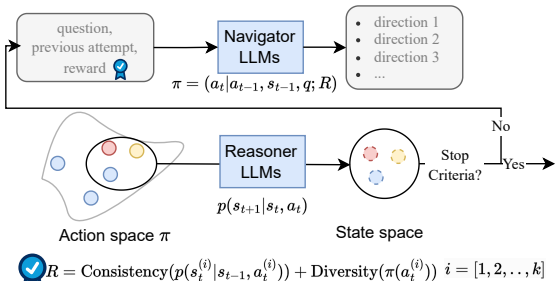


Figure 3: An overview of `Mirror`. It facilitates diverse question-specific directions (represented by different colored dots in the action space) to encourage extensive reflection by the Reasoner. The stopping criterion is based on the consistency among states from multiple perspectives, which also contributes to the direction generation.

spectively in the t-th reflection iteration. In the context of multiple-choice question, a_t is the direction generated by the Navigator, and s_t is the response generated by the Reasoner, including the answer to the question and the rationale behind. $\mathcal{R}(s, a)$ is the reward function. Therefore, we have state transition distribution $\mathcal{P}(s_t|s_{t-1}, a_{t-1})$ and action generation distribution $\pi(a_t|s_t, q, p_0, \mathcal{R})$, where p_0 is the prompt for the Navigator to generate direction a_t. It is nontrivial to obtain frequent rewards that incentivize self-refinement progress without access to the ground truth. Therefore, we turn to an intrinsically motivated planning algorithm, i.e., Monte-Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006; Browne et al., 2012; Swiechowski et al., 2023) to efficiently explore the environment augmenting rewards with auxiliary objectives (Mu et al., 2022; Du et al., 2023b).

Comparing to existing work search-based reasoning methods based on frozen LLMs (Hao et al., 2023; Zhou et al., 2023), we highlight two notable contributions addressing the vulnerabilities of LLMs as discussed in §3: (1) *Step-wise Multiple-perspective self-assessment:* unlike approaches that rely on ground truth or majority-voting based on several complete generated trajectories, our framework utilizes multiple-perspective consistency as stop criteria at each step t. (2) *Novel Reward Mechanism:* a novel diversity mechanism is designed to avoid the null space encountered in traditional random search settings. Our method is detailed in Algorithm 1 in the Appendix.

4.2 Multiple-perspective Assessment

Motivated by the empirical results in § 3.1 regarding knowledge-grounding, we propose to employ an advanced consistency-based method as a surrogate for factual correctness when external resources are unavailable. This method considers both intra- and inter-consistency of the generated responses. Specially, we employ the Navigator for K question-oriented direction generation, $a \sim \pi(a_t|q, s_t, p_0, \mathcal{R})$. These K directions are intended to provide diverse perspectives for problem-solving, with the agreement among guided responses representing inter-consistency. Meanwhile, the confidence in self-consistency (Wang et al., 2023) serves as the measure of intra-consistency. To integrate consistency considerations into the assessment per reflection iteration, we use intra-consistency to determine whether the Reasoner should accept its initial response. If the intra-consistency surpasses a threshold T_0, we consider it as the final result; otherwise, we integrate the inter-consistency as an indicator for stopping criteria in subsequent reflection iterations. We derive the final answer when the inter-consistency exceeds T_0 or when reach the predefined maximum iterations, selecting the final answer with the highest consistency score. This inter-consistency also becomes part of reward $\mathcal{R}_{\text{consistency}}$ for the current state and contribute to the direction generation. We compare with majority voting in Table 4 to illustrate the efficiency of our assessment strategy.

4.3 Diverse and Valid Search Space

Obtaining a meaningful and diverse action space is challenging due to the absence of a dense and well-defined reward function in the planning al-

gorithm. One of the predominant reasons is that different action sequences can lead to similar outcomes (Baranes and Oudeyer, 2013). In our context, considering the limitation of LLMs in following instructions, the Reasoner may ignore the differences among multiple directions and generate identical responses merely based on the question. Therefore, some intrinsically motivated reinforcement learning algorithms choose to explore outcomes rather than actions (Oudeyer and Kaplan, 2007; Ladosz et al., 2022). MCTS addresses the limitation of sparse rewards by visiting novel states or transitions through random exploration (Du et al., 2023b). The most popular algorithm in the MCTS family, Upper Confidence Bound for Trees (UCT) (Kocsis and Szepesvári, 2006) is treated as the choice of child node, UCT $= \overline{\mathcal{R}}_j + 2C_p\sqrt{\frac{2\ln N(n)}{N(n_j)}}$, where $\overline{\mathcal{R}}_j$ is the average reward for child node j, while the second term encourages sampling from nodes whose children are less visited. $N(n)$ is the number of times current node (parent) has been visited in previous iterations, and $N(n_j)$ is times of the child node has been visited. The $C_p > 0$ is a constant to control balance between exploitation (first term) and exploration (second term). In our case, we specifically promote diversity between the parent and child node, i.e., the response in previous attempt s_{t-1} and the current attempt s_t[3]. For multiple-choice questions in MMLU, we assess if the predicted choices are the same across two reflection iterations. The discrepancy in responses indicates the alleviation of null direction space and the avoidance of being stuck, especially given the relatively low consistency with the response from the previous iteration. The relationship between task performance and the diversity of responses in the generated tree, as illustrated in Figure 5, confirms our motivation for diversity enhancement. However, maximizing diversity of outcomes may not always be enough, as less relevant states might be collected (Du et al., 2023b). Therefore, we filter out states whose associated responses are not in the correct form, such as failing to provide a final choice, or refusing to answer questions for moral considerations.

[3]We conducted experiments by filtering K distinct directions based on their semantic similarity derived from Sentence-BERT (Reimers and Gurevych, 2019), but the performances did not match those achieved by directly constraining the diversity of outcomes.

5 Can Mirror Steer LLMs in Iterative Improvements?

We evaluate our proposed `Mirror` on MMLU and FEVER (Thorne et al., 2018). FEVER is a fact-checking dataset featuring three labels for knowledge-rich statements, i.e., `supports`, `refutes` and `not enough info`.

5.1 Experimental Setup and Results

Comparison methods. The evaluation models are GPT-3.5, Llama2-13B-Chat (Touvron et al., 2023), and Vicuna-v1.5-7B (Zheng et al., 2023) [4]. We equip the LLMs with different reasoning mechanisms, including Chain-of-Thought (CoT) (Wei et al., 2022), Self-consistency (Wang et al., 2023), Self-Correction (Huang et al., 2023) and Reflexion(w.GT) (Shinn et al., 2023). We implement CoT by prompting LLMs to first generate step-by-step thoughts and then generate answers based on those thoughts. We repeat this process for five times, resulting in Self-Consistency$^{(5)}$. The remaining two methods are self-improvement techniques where LLMs are first prompted to generate reflections, followed by updating their current response accordingly if applicable. Self-Correction relies on LLM's internal knowledge for answer assessment, while Reflexion compares the current answer with the ground truth for evaluation.

Methods	STEM	Social	Hum	Others	FEVER
Relexion(w.GT)$^{(5)}$	0.79	0.84	0.78	0.73	0.72
GPT-3.5 (CoT)	0.63	0.65	0.53	0.60	0.58
Self-Consistency$^{(5)}$	0.67	0.68	0.58	0.64	0.61
Self-Correct$^{(2)}$	0.63	0.62	0.55	0.54	0.55
`Mirror`	0.76	0.77	0.71	0.67	0.64
Relexion(w.GT)$^{(5)}$	0.64	0.63	0.60	0.64	0.59
Llama13B(CoT)	0.42	0.58	0.42	0.53	0.40
Self-Consistency$^{(5)}$	0.45	0.60	0.49	0.57	0.46
Self-Correct$^{(2)}$	0.42	0.52	0.53	0.45	0.36
`Mirror`	0.57	0.62	0.58	0.62	0.54
Relexion(w.GT)$^{(5)}$	0.62	0.68	0.59	0.69	0.59
Vicuna13B (CoT)	0.46	0.57	0.43	0.57	0.39
Self-Consistency$^{(5)}$	0.50	0.62	0.53	0.60	0.43
Self-Correct$^{(2)}$	0.43	0.49	0.42	0.49	0.38
`Mirror`	0.59	0.64	0.56	0.65	0.46

Table 2: Performances of different reasoning methods, with an upper-bound represented by results obtained when ground truth is provided, denoted as Relexion(w.GT). The superscripts denote the number of reasoning iterations.

Results. The results are shown in Table 2. By comparing CoT with Self-Correction, we observe the performance degradation after two rounds of

[4]We denote them as Llama13B and Vicuna13B for simplicity. Experiment details can be found in Appendix C.

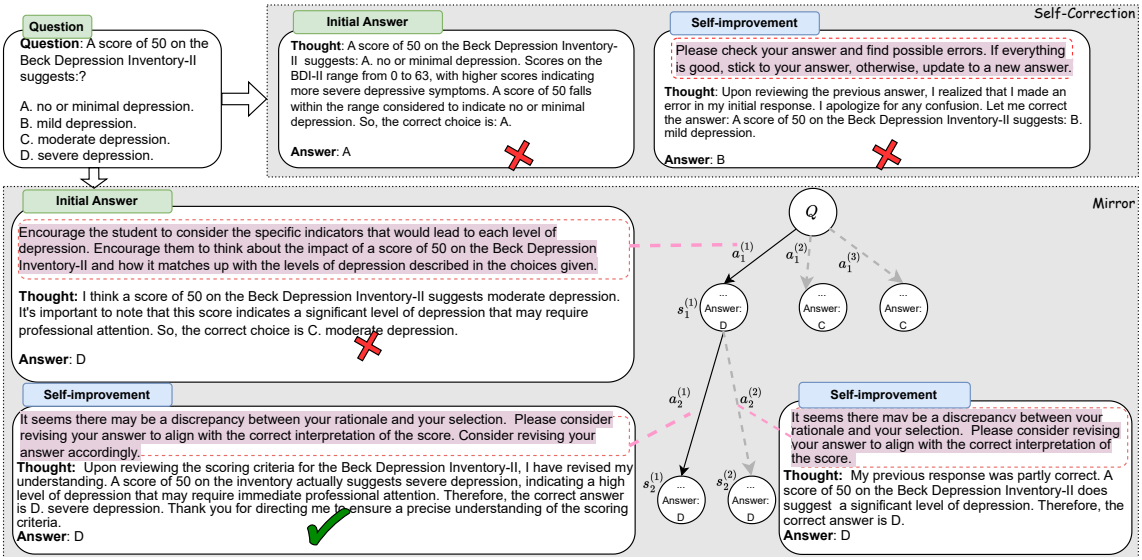


Figure 4: Reasoning process of self-correction and `Mirror`. Text in red are generated directions.

self-Correction across almost all datasets and models. This observation aligns with our findings in §3.1 and in (Huang et al., 2023). Equipped with self-consistency$^{(5)}$, significant performance improvements are evident across all settings. `Mirror` considers additional inter-consistency, achieves the most notable improvements, with a relative increase of more than 15% across the three models. Figure 4 illustrates the reasoning process of Self-correction and `Mirror`. Both methods fail to answer correctly in the first trial. With question-oriented direction, the Reasoner better identify errors in the initial response, such as, *the error in score direction* and *inconsistency between rationales and selection*. The consistency-based criteria built in the tree further improves the fact assessment. During backpropagation, node $s_1^{(1)}$ receives a higher reward, leading to the leftmost reasoning path (details of direction $a_1^{(1)}, a_2^{(1)}, a_2^{(2)}$ and corresponding responses are shown in the text frame). By contrast, Self-correction seems to engage in groundless inference by switching answers without explicit clues. Even comparing `Mirror` with Relexion(w.GT), we find comparable results for GPT-3.5 on the STEM dataset, for Llama on all datasets except for STEM and for Vicuna on STEM and Humanity. From the perspective of the model, the average improvements over baselines for GPT-3.5 are particularly prominent, partly explained by its better ability to adhere to provided directions. This can also explain the marginal improvements even ground truth are accessible to the smaller models.

5.2 Analysis

We discuss the effects of key strategies in `Mirror`.

Question-Oriented Direction. Motivated by the findings in § 3.2 that LLMs struggle to effectively reflect on themselves with generic feedback, `Mirror` is equipped with a Navigator for generating question-oriented directions. To study the effects of these directions (results in Table 3), we adopt our Navigator for direction generation for CoT settings, in which the direction (GenerativeDirect) is introduced before the LLM generates its thought on the previous trial. We then replace all adaptive directions with a single generic direction (FixedDirect) which reads: `Read the question and choices carefully and diagnose the previous response by locating the incorrect clues and update the response if applicable.` Comparing with CoT, the inclusion of GenerativeDirect boosts the performance across all settings with significant improvements. Conversely, FixedDirect sometimes results in performance degradation for Llama13B. The impact of FixedDirect is similar to advanced instruction intended to provide general direction for the task, whereas GenerativeDirect offers question-specific advice to accurately summarize clues for solution. Referencing to the example in Figure A3, `Mirror` (bottom) firstly prompts the Navigator for direction generation (highlighted in red), which captures the key elements, such as *"the characteristics of a connected and undirected graph"*. The Reasoner then follows this direction to explain the key concepts of this graph, laying a solid founda-

tion for reaching the correct conclusion. Without such direction, the Reasoner may overlook or misinterpret knowledge about this graph, leading to errors in the conclusion.

Models	Methods	MMLU	FEVER
GPT-3.5:	CoT	0.68	0.58
	+ FixedDirect	0.73	0.60
	+ GenerativeDirect	0.78	0.64
Llama13B:	CoT	0.46	0.40
	+ FixedDirect	0.43	0.39
	+ GenerativeDirect	0.49	0.45
Vicuna13B:	CoT	0.48	0.42
	+ FixedDirect	0.51	0.43
	+ GenerativeDirect	0.55	0.45

Table 3: Performances of using generic fixed direction and generative direction on top of CoT.

Diversity of the Search Space. We demonstrate the impact of multiple-perspective directions, aiming at guiding the Reasoner out of reflection traps. To this end, we compute the percentage of generated trajectories containing the correct answers (ans_presence) and the according task performances (acc) across various action space sizes, i.e., the number of generated directions. The results in Figure 5 indicate that lager search space enhanced by the $\mathcal{R}_{\text{diversity}}$ can increase the probability of reaching the correct answer.

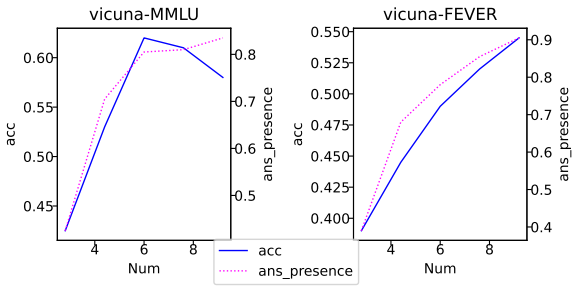


Figure 5: The Accuracy (acc) and the percentage of samples where the ground truth is included in the tree (ans-presence), with different sizes of search space (Num). Results for GPT-3.5 and Llama13B are in Figure A4a and A4b.

Performance of Answer Assessment Criteria. As discussed in Section 3.1, LLMs struggle to assess the correctness of knowledge-rich statements, a capability that can be consistently enhanced through self-consistency. We further reform the majority-voting assessment process by considering the inter-consistency built in the hierarchical decision-making tree. To study the effects of our answer assessment criteria described in §4.2, we compare them with two other voting methods, i.e., self-consistency (majority vote for 5 CoT-generated reasoning trajectories) and majority vote within our generated tree-trajectories. We average the results from Table 2 for CoT and Self-consistency$^{(5)}$ across four domains in MMLU and denote them as CoT$^{(1)}$ and CoT$^{(5)}$, respectively. For Majority$^{(tree)}$, we select the final answer through majority-voting among all intermediate nodes in our generated tree-trajectories. The results of different final answer assessments are presented in Table 4. We observe a performance increase after applying majority-voting in the CoT settings, while this simple strategy doesn't yield improvements in the generated tree. This is because undesirable responses may be generated during the node expanding phase, and majority voting treats all nodes equally. In contrast, our reward-based search tends to focus on reliable nodes with higher confidence in each reflection step, thereby avoiding less desirable nodes.

Models	Ans. Assessment	MMLU	Fever
GPT-3.5:	CoT$^{(1)}$	0.60	0.58
	CoT$^{(5)}$	0.64	0.61
	+Majority$^{(tree)}$	0.69	0.59
	+Reward Search$^{(tree)}$	0.73	0.64
Llama13B:	CoT$^{(1)}$	0.49	0.40
	CoT$^{(5)}$	0.53	0.46
	+Majority$^{(tree)}$	0.58	0.50
	+Reward Search$^{(tree)}$	0.60	0.54
Vicuna13B:	CoT$^{(1)}$	0.51	0.39
	CoT$^{(5)}$	0.56	0.43
	+Majority$^{(tree)}$	0.59	0.43
	+Reward Search$^{(tree)}$	0.60	0.46

Table 4: Results of different answer assessment methods.

6 Conclusion

In this paper, we present a multiple-perspective reflection method, called `Mirror`, for knowledge-enriched reasoning. To tackle the limitations of LLMs in fact assessment and the generation of high-quality feedback, `Mirror` is equipped with a directional Navigator, enabling the Reasoner to identify multiple key clues in problem-solving. Furthermore, the consistency among responses generated under different directions enhances the validity of answer assessment, particularly when ground truth is not accessible. Experiments conducted on five reasoning datasets demonstrate `Mirror`'s superiority over several contemporary CoT-based and self-consistency-based reasoning approaches. Moreover, the ablation study results clearly show that our strategies effectively alleviate the aforementioned challenges.

7 Limitations

In this study, our primary focus on identifying optimal reasoning trajectories based on generated outputs and frozen states. However, the fact-assessment and reflection generation capabilities may be limited by the intact decoding process and pre-training. To fully leverage the potential of LLMs in complex reasoning, it is valuable to explore in the two directions: (1) Strategically guiding the fine-grained generation, such as token-level generation in the decoding phase within the expansive generation space. (2) Fine-tuning LLMs using access to limited task-oriented data to enhance their responses to more complex problems.

8 Ethics Statement

We utilized two publicly available datasets: Massive Multitask Language Understanding (MMLU) and FEVER (Fact Extraction and Verification). MMLU is a multiple-choice question-answering dataset covering 57 subjects across STEM, social sciences, humanities, and more. Notably, some subjects, such as *moral disputes* and *moral scenarios*, contain statements raising ethical concerns. Large language models may pose a risk of misuse or misjudgment in these contexts. We strongly advise thorough consideration of safety implications before applying relevant techniques in real-world scenarios. For the FEVER dataset, positive claims (facts) are extracted from Wikipedia, and negative claims are generated by contrasting these facts and subsequently verified without knowledge of their source sentences. However, considering Wikipedia as a social network where virtually anyone can revise content, the extracted facts may not be perfect. Consequently, we discourage the usage of our work as ground truth for any fact verification task in case any confusion and bias.

References

Adrien Baranes and Pierre-Yves Oudeyer. 2013. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robotics Auton. Syst.*, 61(1):49–73.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. 2012. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. 2023a. Learning universal policies via text-guided video generation. *CoRR*, abs/2302.00111.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023b. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 8657–8677. PMLR.

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16477–16508, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling large language models to generate text with citations. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Amelia Glaese, Nathan McAleese, Maja Trkebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, A. See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Sovna Mokr'a, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William S. Isaac, John F. J. Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements. *ArXiv*, abs/2209.14375.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023a. CRITIC: large language models can self-correct with tool-interactive critiquing. *CoRR*, abs/2305.11738.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2023b. Critic: Large language models can self-correct with tool-interactive critiquing. *ArXiv*, abs/2305.11738.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. *CoRR*, abs/2305.14992.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *CoRR*, abs/2310.01798.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Ho Hin Lee, and Lu Wang. 2023. Grace: Discriminator-guided chain-of-thought reasoning. In *Conference on Empirical Methods in Natural Language Processing*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Pawel Ladosz, Lilian Weng, Minwoo Kim, and Hyondong Oh. 2022. Exploration in deep reinforcement learning: A survey. *Information Fusion*, 85:1–22.

Cheng Li, Jindong Wang, Kaijie Zhu, Yixuan Zhang, Wenxin Hou, Jianxun Lian, and Xing Xie. 2023. Emotionprompt: Leveraging psychology for large language models enhancement via emotional stimulus. *arXiv preprint arXiv:2307.11760*.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023a. Lost in the middle: How language models use long contexts. *CoRR*, abs/2307.03172.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M Dai, Diyi Yang, and Soroush Vosoughi. 2023b. Training socially aligned language models in simulated human society. *arXiv preprint arXiv:2305.16960*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *CoRR*, abs/2303.17651.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *CoRR*, abs/2303.08896.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *ArXiv*, abs/2310.06824.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 11048–11064. Association for Computational Linguistics.

Jesse Mu, Victor Zhong, Roberta Raileanu, Minqi Jiang, Noah Goodman, Tim Rocktäschel, and Edward Grefenstette. 2022. Improving intrinsic exploration with language abstractions. *Advances in Neural Information Processing Systems*, 35:33947–33960.

Pierre-Yves Oudeyer and Frederic Kaplan. 2007. What is intrinsic motivation? a typology of computational approaches. *Frontiers in neurorobotics*, 1:6.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Dinesh Parthasarathy, Georgios D. Kontes, Axel Plinge, and Christopher Mutschler. 2023. C-MCTS: safe planning with monte carlo tree search. *CoRR*, abs/2305.16209.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *ArXiv*, abs/2304.01904.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, and Jianfeng Gao. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *CoRR*, abs/2302.12813.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Conference on Empirical Methods in Natural Language Processing*.

Noah Shinn, Federico Cassano, Beck Labash, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning.

Maciej Swiechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mandziuk. 2023. Monte carlo tree search: a review of recent modifications and applications. *Artif. Intell. Rev.*, 56(3):2497–2562.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *NAACL-HLT*.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, MingSung Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STar: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.

Tianhua Zhang, Hongyin Luo, Yung-Sung Chuang, Wei Fang, Luc Gaitskell, Thomas Hartvigsen, Xixin Wu, Danny Fox, Helen M. Meng, and James R. Glass. 2023. Interpretable unified language checking. *ArXiv*, abs/2304.03728.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2023. Language agent tree search unifies reasoning acting and planning in language models. *CoRR*, abs/2310.04406.

Xinyu Zhu, Junjie Wang, Lin Zhang, Yuxiang Zhang, Yongfeng Huang, Ruyi Gan, Jiaxing Zhang, and Yujiu Yang. 2023. Solving math word problems via cooperative reasoning induced language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4471–4485, Toronto, Canada. Association for Computational Linguistics.

A More Experimental Details for Initial Study

A.1 Experiment for Figure 1

The prompt used in *Autostop* is "`You were either successful or unsuccessful in your previous trial. Stick to your previous answer if it is correct, otherwise consider a new answer`". The prompt used for *NeverStop* is "`You failed in your previous trial and reconsider a new answer`". The motivation behind *Autostop* is that we totally rely on the LLM's internal knowledge to check the correctness of its own outputs. However, LLM fails in this setting as the performance is even worse than initial stage. For *NeverStop*, we hope to identify that some correctly answered samples will be kept unchanged even the negative feedback provided. However, we didn't find a pattern between the changed and unchanged predicted samples.

A.2 Implementation for Knowledge Grounding and Results

Dataset We evaluate LLMs' knowledge grounding ability on knowledge-rich multiple-choice dataset, MMLU. It consists of four domains: STEM, Social, Humanity and Other, totaling 56 subjects. All methods are evaluated on 50 randomly selected samples for each subject (excluding those in the Other domain), and the remaining samples are used as the training set where applicable.

Models and Baselines In addition to Llama2-13B, Llama2-70B, and GPT-3.5 for prompting, we also leverage unified language checking, *UniLangCheck* (Zhang et al., 2023), for statement assessment. *UniLangCheck* aims to check if language input is factual and fair via prompting LLMs to generate groundings for fact-checking. Therefore, we firstly prompt LLMs to generate a fact about the key element in the question before proceeding to the final assessment. We repeatedly prompt the LLMs for 5 times and use the majority-voted answer as the result for *Self-Consistency* (Wang et al., 2023). *TRUE* (Honovich et al., 2022) is the T5-11B (Raffel et al., 2020) model fine-tuned on a collection of natural language inference (NLI) datasets to check factual correctness, and has been used by previous works within similar contexts (Gao et al., 2023a,b). We further fine-tune its classifier head on our training set, which is annotated as factually correct or not, before evaluation. Both Contrastive Consistent Search (*ContrastSearch*) (Burns et al., 2023) and *ActivationRegress* (Marks and Tegmark, 2023) train classifiers whose inputs are activations extracted from Llama2-13B 12-layer encodings [5]. *ActivationRegress* trains a logistic classifier on the activations with factual labels as supervision. *ContrastSearch*, instead, operates without factual labels. For a statement s_i, we firstly construct a data-pair x^+ and x^- by annotating *True* and *False* to this statement ,regardless of its factual correctness. Then, we derive the probabilities by mapping x to a number between 0 and 1, i.e., $p^+ = p_\theta(\phi(x_i^+))$ and $p^- = p_\theta(\phi(x_i^-))$. The mapping function p_θ is updated such that the probabilities are both confident ($p_i^+ \approx 1 - p_i^-$) and consistent ($p_i^+ \not\approx p_i^-$).

Prompt Settings The basic prompt for knowledge grounding is shown in Figure A1a. This is used for Llama2, GPT-3.5 and Self-Consistency. The advanced prompt inspired by *UniLangCheck* is illustrated in Figure A1b. For each subject, we randomly select 50 samples and extract their question and choice to build a statement for knowledge checking. The correctness of this statement is deemed *True* if the selected choice is exactly the correct one, otherwise it is labeled *False*.

Correlation between Self-consistency Confidence and Accuracy For the self-consistency(5) baseline, we calculate the $\mathcal{R}^2$ for confidence (the frequency of the current answer among all generated answers, totaling 5) and the accuracy. The results are shown in Table A1. We observe a high correlation between the two variables, which inspires our design of multiple-consistency for answer assessment.

	STEM	Social	Humanity	Others
GPT-3.5	0.80	0.89	0.84	0.88
Llama	0.86	0.85	0.91	0.86
Vicuna	0.92	0.90	0.74	0.92

Table A1: The correlation between accuracy and self-consistency confidence over three domains in the MMLU datasets.

A.3 Implementation for Direction Generation

Based on the observation that existing feedback has limited effects to guide LLMs to update their

[5] The original dimensions of Llama2-30B is 5024. We apply PCA to reduce this dimensionality to obtain 50-dimensional activations as classifier input.

As an expert in knowledge grounding, you'll be assessing statements that consist of a question followed by a proposed answer. The question forms the initial part of the statement, and the answer follows it. Utilize your thoughtful analysis to determine the correctness of each statement. Conclude the assessment with a "Finish[answer]" that returns either True or False, marking the completion of the task.

Here are some examples:
{examples}
(END OF EXAMPLES)

Statement: {question:q, answer: a}
Thought: thought
Action: Finish[answer]

(a) Basic prompt for knowledge grounding. Text in gray is extracted from datasets, in red shadow is generated by LLMs.

In your capacity as a specialist in knowledge grounding, your task is streamlined into a comprehensible two-step process. Firstly, assume the role of a question architect, delineating the essential "key elements/knowledge" integral to formulating a sound question. Subsequently, based on these identified key knowledge elements, proffer your response. The ensuing step involves a meticulous comparison of your proposed answer with the provided solution to ascertain accuracy. Conclude this evaluative process with a succinct 'Finish[answer]' statement, conclusively designating either True or False, thereby encapsulating the successful execution of the task."
Here are some examples{examples}
(END OF EXAMPLES)

Statement: {question:q, answer: a}
Key Element/fact: fact
Thought: thought
Comparison: comparison
Action: Finish[answer]

(b) Fact-extract prompt applied to *UniLangCheck* for knowledge grounding. Text in gray shadow is extracted from datasets, in red shadow is generated by LLMs. Comparing to the basic prompt, it includes additional fact generation.

current incorrect response, we propose several simple strategies to enhance the effectiveness of generated feedback in the self-improvement process. These strategies are mainly inspired by the following two observations: (1) LLMs are more susceptible to context influence at the beginning or near the end (Liu et al., 2023a) (2) ICL is highly sensitive to the stylish and emotional words in demonstrations (Min et al., 2022; Li et al., 2023). We summarize the different strategies in the diagram shown in Figure A2.

We show the performances over three LLMs after applying different instructions in Table A3. It is clear that *NegPrefix* demonstrates the most significant improvements across all the datasets and models. In contrast, *NewAnswer* has the same sentences *NegPrefix* as but its position is far away from the generating point for LLMs. *This can be explained that position of instruction is important in ICL.* And the performance of *NewAnswer* is slightly better than baseline, it can be partly explained that the *NewAnswer* explicitly show the negative attitude towards and guide the model to generate a different answer. Among the three models, the average promotion on GPT3.5 is the most negligible. *This can be explained that larger model are more confident with its internal knowledge and less vulnerable to given noisy text.*

Model	**Prompts**	Change
GPT35	Oracle	0.56
	NegReflect	0.72
Llama	Oracle	0.54
	NegReflect	0.72
Vicuna	Oracle	0.64
	NegReflect	0.74

Table A2: The relative percentage of changed samples among those incorrectly predicted ones. We use the average results for different domains in MMLU.

B Mirror algorithm

We introduce the pipeline of the proposed Mirror in Algorithm 1 involves iteratively conducting a UCT-SEARCH until predefined iteration constraint is reached, and the best action $a(\text{BESTCHILD}(v_0, 0))$ leading to the best child of the root node v_0 returns. Node in the tree is v and its associated state is $s(v)$, representing the response generated by Reasoner. The action is $a(v)$, reward is $\mathcal{R}$ and $N(\cdot)$ is the times of the node having been visited. $r(v)$ is the reward for the terminate state at each iteration.

The overall process consists of three steps: (1) SEARCHPOLICY to obtain the terminal node v_l. through which expands the tree until fully expanded. Specially, we randomly add one or more nodes to the root node according to the possible ac-

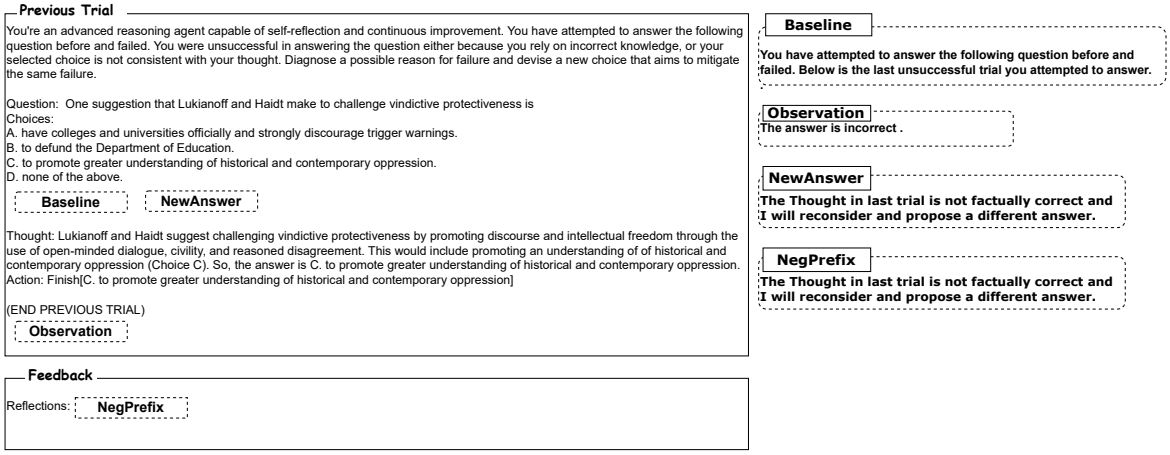


Figure A2: Given the question and the LLM's *previous trial*, it is asked to generated *feedback* under different prompts to facilitate reflection and potentially update its previous response. The four candidate instructions, Baseline, Observation, NewAnswer and NegPrefix, are enclosed in dashed frames, and they will be positioned differently to exert their respective effects.

Model	Stem	Social	Humanity	Other
GPT35	0.80	0.82	0.78	0.73
+Observation	0.76	0.82	0.75	0.70
+NewAnswer	0.80	0.84	0.80	0.75
+NegReflect	0.84	0.86	0.84	0.76
Llama	0.64	0.63	0.60	0.64
+Observation	0.63	0.62	0.61	0.62
+NewAnswer	0.64	0.67	0.65	0.64
+NegReflect	0.70	0.72	0.76	0.69
Vicuna	0.62	0.68	0.59	0.69
+Observation	0.64	0.52	0.45	0.67
+NewAnswer	0.66	0.58	0.47	0.65
+NegReflect	0.69	0.63	0.52	0.72

Table A3: Self-improvments results with different prompt constraints for answer correction. By comparing with the ground truth, this evaluation is to show the capability of LLMs in obeying the instructions of changing their incorrect predictions.

tions. In our case, we generate multiple responses to the given question and previous attempts/response. When the current node is fully expanded, we apply the UTC algorithm to select the best child node. (2) SIMULATION the reward r for v_l through SIMULATIONPOLICY. This phrase is to simulate the future rewards of the current node through multiple interactions. For simplicity, we follow the similar process as expansion and return the reward r for selected action-state pair. (3) BACKPROPAGATE the simulation results to the selected nodes to accelerate SEARCHPOLICY in next iteration.

C Experiments for Mirror

We will introduce the implementation details and provide complementary results experimented on `Mirror` in this section.

C.1 Implementation Details

Hyper-parameter settings. In order to encourage diverse direction generation, we set the generation temperature as 0.8 for all the models, and we set `do_Sample` = `True` for llama and vicuna to avoid greedy search. For the threshold T_0 in self-assessment to deriving the final answer, we set 0.8 for GPT35, and 0.5 for llama and Vicuna according to the results on limited validation data.

Prompt Settings. We provide 5 demonstrations along with instruction when prompting LLMs. We show the prompts/instructions provided to LLMs in direction generation and response generation process.

(a) p_0 in direction generation in $\pi(a_t|s_t, p_0, \mathcal{R})$. The guidance in the upper is for initial response, the bottom one is for reflection in the subsequent iterations.

(b) Prompt for response generation given previous response and direction. $\mathcal{P}(s_t|s_{t-1}, a_{t-1}; q)$

Prompt for Direction Generation (MMLU)

As a tutor, your focus is on guiding the student to navigate multiple-choice question-answering problems strategically. Encourage them to dissect the question, identifying key elements and nuances within each choice. Emphasize the importance of understanding subtle differences that could distinguish correct from incorrect options.

As a tutor, your are supposed to meticulously evaluate the student's approach to multiple-choice problems. Question, Choices and the student's previous thought and answer are given, check if the facts mentioned in the thought is correct and if there might be a more appropriate option than the one chosen. If the student's reasoning thought is accurate and the proposed answer is the most appropriate, encourage them to adhere to their initial trial. Otherwise, guide the student to revisit specific details, explore alternative choice.

Prompt for Direction Generation (FEVER)

As a tutor, your focus is on guiding the student to navigate fact-checking problems strategically. Encourage them to dissect the claim, identifying key elements and associate facts. Emphasize the correct relation between important elements that could distinguish SUPPORTS from REFUTES options. Also, lacking of enough information will lead to NOT ENOUGH INFO.

As a tutor, your are supposed to meticulously evaluate the student's approach to fact verification task. Claim and the student's previous thought and answer are given, check if the relations mentioned in the Thought is correct and if there might be a more appropriate answer. If the student's reasoning thought is accurate and the proposed answer is the most appropriate, encourage them to adhere to their initial trial. Otherwise, guide the student to revisit specific details, explore alternative answer.

Prompt for Response Generation (MMLU)

You are an expert in multiple-choice question answering. Each problem will provide you with a question and answer choices. Read the question and all the choices carefully, along with the provided advice, and solve the problem by having a thought. Thought can reason about the current situation. Finish[answer] returns the answer and finishes the task.

You're an advanced reasoning agent capable of self-reflection and continuous improvement. Your objective is to tackle multiple-choice question answering problems. Each problem will provide you with a question, answer choices, your previous line of reasoning, and the detailed analyses from an experienced tutor. In a succinct review, assess the accuracy of your earlier answer based on your expertise and the advice, subsequently arrive at the definitive response.

Prompt for Response Generation (FEVER)

You are a knowledgeable and accurate fact verifier. Please verify the correctness of the following claim based on your expertise and provided advice. Return SUPPORTS or REFUTES a Claim, or if there is NOT ENOUGH INFO.

You're an advanced reasoning agent capable of self-reflection in fact verification task. Claim and the your previous response and answer are given, along with the advice. In a succinct review, assess the accuracy of your earlier answer based on your expertise and the advice, subsequently arrive at the definitive response.

Computational budget. The total running costs for using GPT-3.5 in our experiments are approximately $500. In addition, the running time for Llama2 and Vicuna in our experiments totalled 320 hours, utilising one 40G A100 graphics cards.

C.2 Additional Results

Effects of question-oriented direction. To save computational resources, we randomly select 20 samples from each of the four domain datasets in MMLU and from FEVER. We show an example of generated direction in the CoT settings.

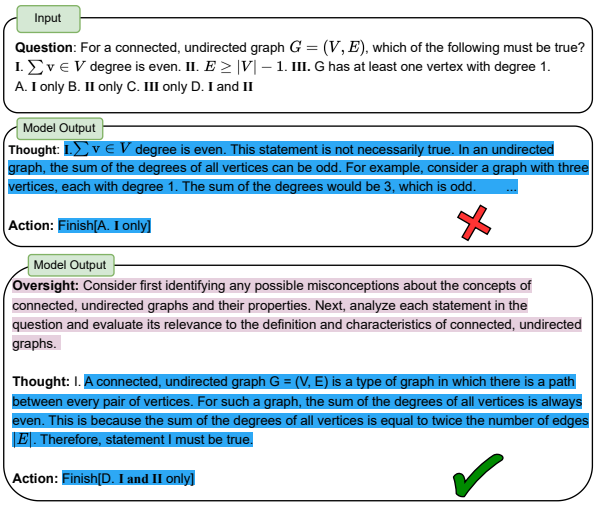


Figure A3: With question-oriented direction, the Reasoner answers questions with explicit clues.

The diversity of Search Space One of our motivations is to broaden the diversity of actions available for more effective exploration. Consequently, we compute the upper bound results for our generated tree, indicating the presence of the correct answer in the tree signifies a correctly answered sample. Results are shown in Figure 5.

Algorithm 1 `Mirror-UCT`

Require: state transition function $f : S \times A \to S$, weight C_p, Reward $\mathcal{R}$, Stop Criteria $g \to \{0, 1\}$

function UCT-SEARCH(s_0)
  create root node v_0 with state s_0
  while within computational iteration **do**
    $v_l \leftarrow$ SEARCHPOLICY(v_0)
    $r \leftarrow$ SIMULATIONPOLICY(s_{v_l})
    BACKPROPAGATE(v_l, r)
  return a(BESTCHILD($v_0, 0$))

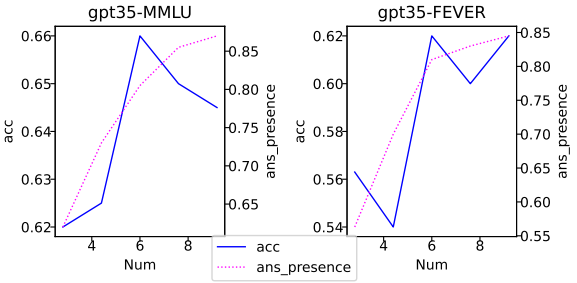
function SEARCHPOLICY(v)
  while $g(v) == 0$ **do**
    if v not fully expanded **then**
      return EXPAND(v)
    else
      $v \leftarrow$ BESTCHILD(v, C_p)
    return v

function EXPAND(v)
  choose $a \in$ untried actions from $A(s(v))$
  add a new child v' to v
    with $s(v)' = f(s(v), a)$ and $a(v') = a$
  return v'

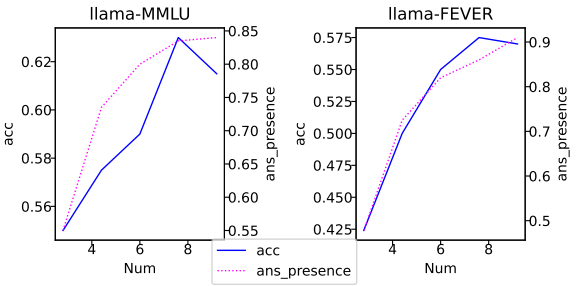
function BESTCHILD(v, C_p)
  return $\underset{v' \in \text{children of } v}{\text{argmax}} \frac{\mathcal{R}_{v'}}{N(v')} + 2C_p\sqrt{\frac{2 \ln N(v)}{N(v')}}$

function SIMULATIONPOLICY(s)
  While s is non-terminal **do**
    $a = \underset{a \in A}{\text{argmax}}(\mathcal{R}(a, s))$
    $s \leftarrow f(s, a)$
  return reward for s

function BACKPROPAGATE(v, r)
  while v is not null **do**
    $N(v) \leftarrow N(v) + 1$
    $\mathcal{R}(v) \leftarrow \mathcal{R}(v) + r(v)$
    $v \leftarrow$ parent of v



(a)



(b)

Figure A4: The task performance, Accuracy (acc) and the percentage of samples where the ground truth is included in the tree (ans-presence), with different size of search space (Num).