When Can LLMs *Actually* Correct Their Own Mistakes? A Critical Survey of Self-Correction of LLMs

Ryo Kamoi[1] **Yusen Zhang**[1] **Nan Zhang**[1] **Jiawei Han**[2] **Rui Zhang**[1]
[1]Penn State University, USA [2]University of Illinois Urbana-Champaign, USA
{ryokamoi,rmz5227}@psu.edu

Abstract

Self-correction is an approach to improving responses from large language models (LLMs) by refining the responses using LLMs during inference. Prior work has proposed various self-correction frameworks using different sources of feedback, including self-evaluation and external feedback. However, there is still no consensus on the question of *when LLMs can correct their own mistakes*, as recent studies also report negative results. In this work, we critically survey broad papers and discuss the conditions required for successful self-correction. We first find that prior studies often do not define their research questions in detail and involve impractical frameworks or unfair evaluations that over-evaluate self-correction. To tackle these issues, we categorize research questions in self-correction research and provide a checklist for designing appropriate experiments. Our critical survey based on the newly categorized research questions shows that (1) no prior work demonstrates successful self-correction with feedback from prompted LLMs, except for studies in tasks that are exceptionally suited for self-correction, (2) self-correction works well in tasks that can use reliable external feedback, and (3) large-scale fine-tuning enables self-correction.

1 Introduction

Self-correction is a popular approach to improve responses from large language models (LLMs) by refining them using LLMs during inference (Bai et al., 2022; Madaan et al., 2023). Extensive studies on self-correction have been conducted in various tasks, including arithmetic reasoning, code generation, and question answering (Gao et al., 2023; Shinn et al., 2023). The simplest approach of self-correction prompts LLMs to provide feedback on their own responses and refine the responses using the feedback (Huang et al., 2024a), under the hypothesis that *recognizing errors is easier than avoiding them* (Saunders et al., 2022). As in Figure 1, self-correction has also been studied using additional information for improving feedback, including external tools such as code interpreters (Chen et al., 2024d; Gou et al., 2024), external knowledge retrieved via web search (Gao et al., 2023; Jiang et al., 2023b), or fine-tuning (Welleck et al., 2023; Ye et al., 2023). However, recent studies also report negative results indicating that LLMs cannot self-correct (Huang et al., 2024a; Gou et al., 2024; Li et al., 2024b) or even self-detect (Chen and Shu, 2024; Tyen et al., 2024; Hong et al., 2024; Jiang et al., 2024; Kamoi et al., 2024) their own mistakes at least in certain conditions. These conflicting observations indicate that further analysis of self-correction is needed.

In this work, we provide a critical survey to investigate the conditions required for successful self-correction. First, our analysis finds that prior studies often do not define their research questions in detail. As a result, many papers fail to provide appropriate experiments to evaluate the research questions they implicitly target. To address this issue, we categorize research questions in self-correction research (§3.1) and discuss frameworks that should be used for verifying each research question (§3.2). Finally, we provide a checklist for designing appropriate experiments (§8).

Next, we analyze prior work to identify when LLMs can self-correct their mistakes, using the new definitions of the research questions. Our analysis highlights that the bottleneck is in the feedback generation (§7). Specifically, (1) no prior work shows successful self-correction with feedback from prompted LLMs in general tasks (§4), (2) self-correction works well in tasks where reliable external feedback is available (§5.1), (3) large-scale fine-tuning enables self-correction (§5.2), and (4) some tasks have properties exceptionally suitable for self-correction (§4). In

Transactions of the Association for Computational Linguistics, vol. 12, pp. 1417–1440, 2024. https://doi.org/10.1162/tacl_a_00713
Action Editor: Grzegorz Chrupała. Submission batch: 5/2024; Revision batch: 8/2024; Published 11/2024.

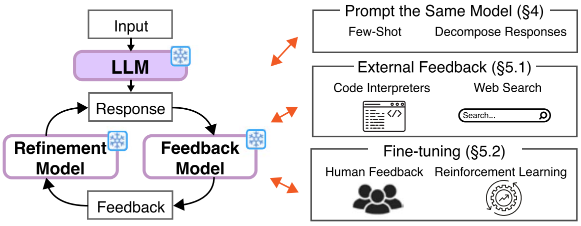

Figure 1: Self-correction in three stages: initial response generation, feedback, and refinement.

summary, our analysis identifies the properties required for successful self-correction as follows:

[RQ1] When can LLMs self-correct *based solely on the inherent capabilities of LLMs?*

- In general tasks, no prior work shows reliable evidence of successful self-correction with in-context learning. (§4)
- In tasks with specific properties that are exceptionally favorable for self-correction (e.g., responses are decomposable), self-correction is effective even with in-context learning. (§4)

[RQ2] When can LLMs self-correct the best-possible initial responses *with external information?*

- Self-correction is effective in tasks where reliable external feedback is available. (§5.1)
- Fine-tuning enables self-correction when large training data is available but is unexplored for small training data. (§5.2)

[RQ3] When are the final outputs of self-correction *better than other approaches*?

- Self-correction is often not compared with sufficiently strong baselines, and it is still unclear whether it is better than other approaches. (§6)

This survey is organized as follows. Section 2 provides an overview of self-correction. Section 3 introduces a new approach to classify research questions and frameworks in self-correction research. Sections 4 and 5 analyze prior work in self-correction with in-context learning and external information (external tools, external knowledge, fine-tuning), respectively. Section 6 explains related approaches that should be compared with self-correction as baselines. Section 7 summarizes our findings from the analysis. Section 8 provides a checklist for self-correction research. Section 9 explains differences from other surveys. Section 10 provides studies related to self-correction. Section 11 provides future directions.

2 Self-Correction of LLMs

The term ‘‘self-correction’’ is used in a wide range of scenarios, from a strict definition in which LLMs refine their own responses by themselves (Madaan et al., 2023; Huang et al., 2024a) to broader concepts that also involve feedback from external tools or knowledge (Shinn et al., 2023; Gou et al., 2024). In this work, we define self-correction as a framework that *refines* responses from LLMs using LLMs *during inference*, possibly with external tools or knowledge. As in Table 1, Figure 2, and Figure 3, self-correction has been studied in various frameworks with different sources of feedback.

2.1 Frameworks

Prior studies propose self-correction frameworks with various different architectures.

Explicit Feedback vs. Direct Refinement. Self-correction often consists of three stages including *feedback generation* (Kim et al., 2023; Madaan et al., 2023; Shinn et al., 2023; Huang et al., 2024a):

- **Initial Response Generation** is a stage of generating initial responses from an LLM.
- **Feedback** model generates feedback given the original input and initial response. This stage may use external tools or knowledge.
- **Refinement** model generates a refined response, given the input, initial response, and feedback.

Direct refinement is another approach that refines responses without generating feedback explicitly (Saunders et al., 2022; Bai et al., 2022; Welleck et al., 2023; Akyurek et al., 2023).

Post-hoc vs. Generation-time. *Post-hoc correction* refines responses after they are generated (Pan et al., 2024). *Generation-time correction* or step-level correction (Paul et al., 2024; Jiang et al., 2023b) improves step-by-step reasoning by providing feedback on intermediate reasoning steps.

Paper	Category	Main Models	Additional Feedback			Main Tasks				
			Oracle	External Tools	Fine-Tuning	Reasoning, Coding	Closed-book, Knowledge	Open-book, Context-based	Open-ended Text Gen	Decom-posable
Self-Correction with **In-context Learning** (*Intrinsic Self-Correction*)										
CoVe (2023)	Intrinsic	PaLM 540B	–	–	–	–	–	–	–	Multiple Answers
CAI Revisions (2022)♠	Intrinsic	52B (no details)	–	–	–	–	–	–	Detoxification	–
Self-Refine (2023)♠	Intrinsic	GPT-3.5, GPT-4	–	–	–	Math, Coding	–	Dialogue	–	–
RCI (2023, §3.1)	Oracle	GPT-3.5-T	✓	–	–	Computer Tasks	CSQA	–	–	–
Reflexion (2023, §4.2)	Oracle	GPT-4	✓	–	–	–	–	HotpotQA (GT Context)	–	–
Self-Correction with **External Tools or Knowledge**										
Reflexion (2023, §4.1, 4.3)	Fair-Asym.	GPT-4	–	Game Envs, Interpreter	–	Games, Coding	–	–	–	–
Self-Debug (2024d)	Fair-Asym.	GPT-3.5-T, GPT-4	–	Code Interpreter	–	Text-to-Code	–	–	–	–
CRITIC (2024)	Fair-Asym.	GPT-3, Llama 2 70B	–	Interpreter, Web Search	–	GSM8k, SVAMP	HotpotQA	–	Detoxification	–
RARR (2023)	Unfair-Asym.	Palm 540B	–	Web Search	–	–	NQ, SQA, QReCC	–	–	–
Reflexion (2023, §4.2)	Oracle	GPT-4	✓	Wikipedia API	–	–	HotpotQA	–	–	–
Self-Correction with **Fine-tuning**										
Self-Critique (2022)	Fair-Asym.	InstructGPT	–	–	Human Assessment	–	–	Topic-based Summarization	–	–
SelFee (2023)	Fair-Asym.	Llama 7B, 13B	–	–	ChatGPT Assessment	MT-Bench	MT-Bench	MT-Bench	MT-Bench	–
Baldur (2023)	Fair-Asym.	Minerva 8B, 62B	–	Proof Assistant	GT Answer	Proof Generation	–	–	–	–
REFINER (2024)	Cross-Model	GPT-3.5 (FB:T5-base)	–	–	Synthetic Data	Math, Logic	–	–	Moral Stories	–
RL4F (2023)	Cross-Model	GPT-3 (FB: T5-large)	–	–	Reinforcement Learning	Action Planning	–	Topic-based Summarization	–	–
Self-Correction (2023, §3.4)	Cross-Model	GPT-3 (FB: GPT-Neo)	–	–	GT Answer, External	GSM8k, SVAMP	–	–	Detoxification	–
Self-Correction (2023, §3.1-3.3)	Unfair-Asym.	GPT-Neo 1.3B, GPT-2	–	–	GT Answer, External	GSM8k, SVAMP	–	–	Detoxification, Const Gen	–
Negative Results of Self-Correction (i.e., LLMs cannot Self-Correct)										
RCI (Table 17) (2023)	Intrinsic	GPT-3.5-T	–	–	–	Computer Tasks	CSQA	–	–	–
CRITIC w/o Tool (2024)	Intrinsic	GPT-3, Llama 2 70B	–	–	–	GSM8k, SVAMP	Closed-book HotpotQA	–	Detoxification	–
Huang et al. (2024a)	Intrinsic	GPT-4-T, GPT-3.5-T	–	–	–	GSM8k	CSQA, HotpotQA	–	–	–

Table 1: Representative studies in self-correction of LLMs. Gray color represents unrealistic settings. ♠: Weak prompts for generating initial responses. FB: Feedback models for cross-model correction.

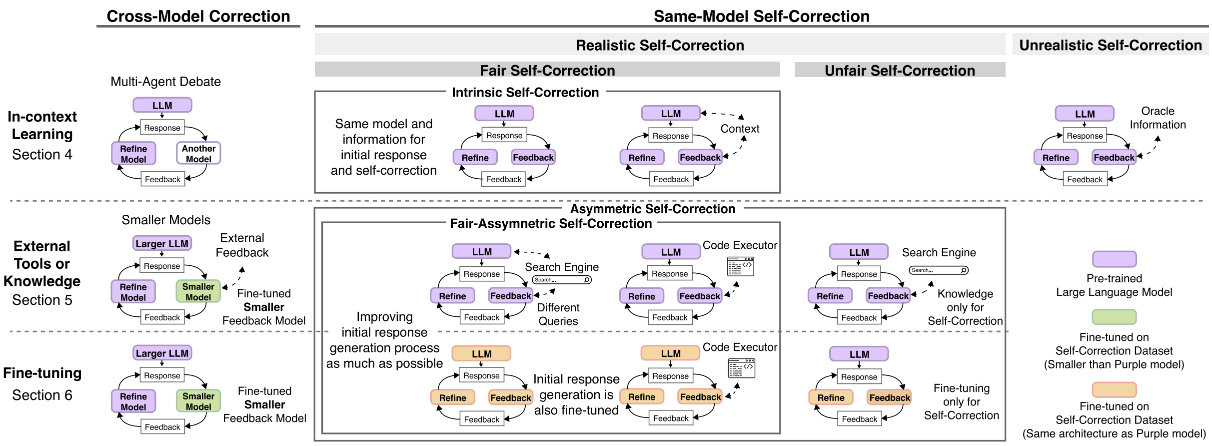

Figure 2: LLM self-correction frameworks, categorized by information used for generating feedback and whether they use best-possible initial responses (§3.2). This figure illustrates representative architectures.

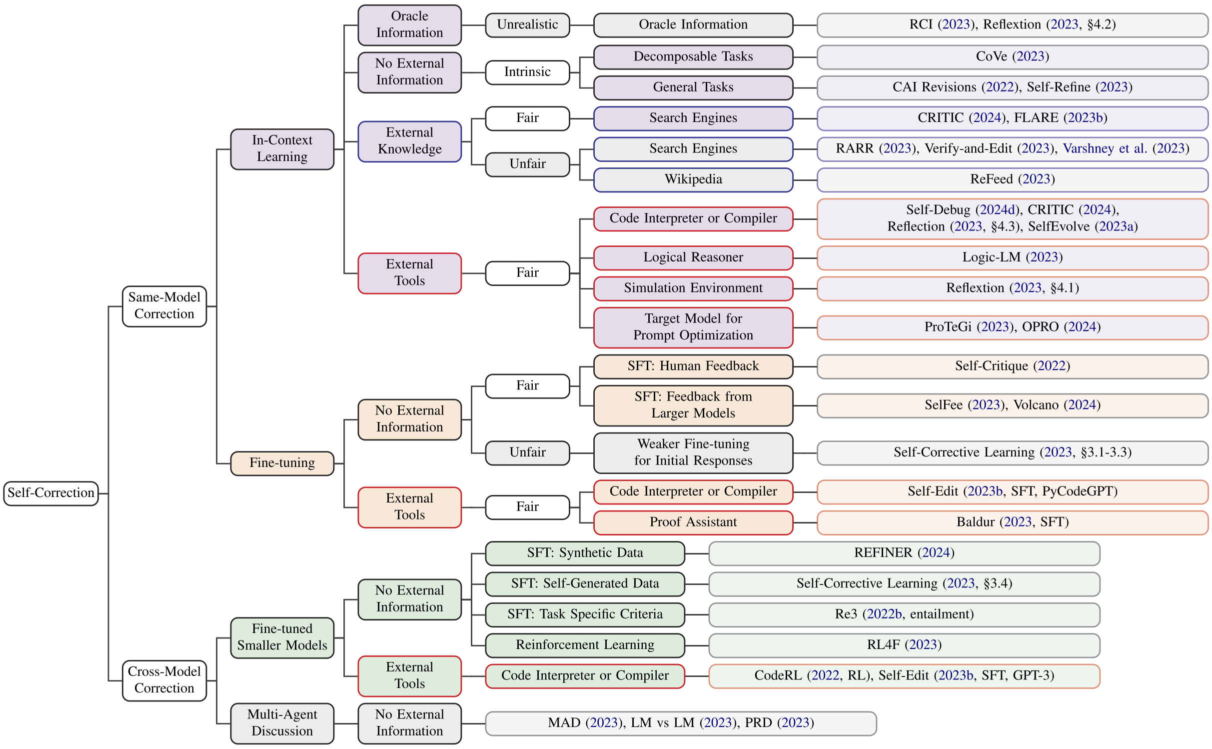

Figure 3: Taxonomy of LLM self-correction, categorized by information used for generating feedback and whether they use best-possible initial responses (fair or unfair). Refer to Section 3.2 for the definitions.

Post-hoc correction is more flexible and applicable to broader tasks, although generation-time correction is popular for reasoning tasks (Pan et al., 2024).

Same-model vs. Cross-model. *Cross-model correction* generates feedback or refines the responses using models different from the model that generates initial responses. Cross-model correction has been mostly studied in the settings of correcting mistakes of large proprietary LLMs using small fine-tuned models (Welleck et al., 2023; Akyurek et al., 2023; Paul et al., 2024) or multi-agent debate of multiple models with similar capabilities (Liang et al., 2023; Li et al., 2023; Cohen et al., 2023; Du et al., 2023; Zhang et al., 2023a; Chen et al., 2024b; Chan et al., 2024; Wang et al., 2024a).

2.2 Sources of Feedback

Intrinsic (§4). Intrinsic self-correction prompts LLMs to generate feedback on their own responses. Prompting strategies include simple zero-shot or few-shot prompts (Madaan et al., 2023; Kim et al., 2023), decomposing the responses (Dhuliawala et al., 2023), and evaluating confidence (Varshney et al., 2023; Jiang et al., 2023b; Wu et al., 2024).

External Information (§5.1). Self-correction often relies on external information, including **external tools** such as code executors (Jiang et al., 2023a; Gou et al., 2024; Chen et al., 2024d; Stengel-Eskin et al., 2024), symbolic reasoners (Pan et al., 2023), proof assistant (First et al., 2023), or task-specific metrics (Xu et al., 2023), **external knowledge** from search engines (Jiang et al., 2023b; Gao et al., 2023; Zhao et al., 2023), Wikipedia (Yu et al., 2023; Zhao et al., 2023), or other corpora (Peng et al., 2023; Zhao et al., 2023), **oracle information** such as ground-truth answers (Kim et al., 2023; Shinn et al., 2023), human feedback (Chen et al., 2024a), or stronger models (Zhang et al., 2024).

Fine-tuning (§5.2). Models fine-tuned for self-correction are another source of feedback, which are trained via supervised fine-tuning (Welleck et al., 2023; Ye et al., 2023; First et al., 2023; Paul et al., 2024; Han et al., 2024) or reinforcement learning (Le et al., 2022; Akyurek et al., 2023).

RQ	Self-Refine (2023)	Huang et al. (2024a)	RCI (2023, §3.1)	RCI (2023, §3.2)	CRITIC (2024, §4.2)	CRITIC (2024, §4.3)	RARR (2023)
RQ1	✓	✗ (§3,5)	✓	–	✗	✗	–
RQ2	–	–	–	✓	✓	✓	–
RQ3	–	✗ (§4)	–	✓	–	✓	✓

Table 2: Research questions that prior studies implicitly target by claiming they are ✓ verified or ✗ refuted.

RQ	Requirements for Frameworks			Required Experiments	
	Information Symmetricity	Best-possible Initial Responses	Realistic	Comparison to Initial Responses	Comparison to Strong Baselines
RQ1	✓	✓	✓	✓	–
RQ2	–	✓	✓	✓	–
RQ3	–	–	✓	–	✓

Table 3: Requirements for experiments to verify each research question in Section 3.1.

2.3 Tasks

Self-correction has been studied in various tasks, including **Reasoning:** arithmetic reasoning (Madaan et al., 2023; Nathani et al., 2023; Gou et al., 2024), code generation (Jiang et al., 2023a; Charalambous et al., 2023; Gou et al., 2024; Chen et al., 2024d; Olausson et al., 2024), proof generation (First et al., 2023), logical reasoning (Pan et al., 2023); **Knowledge:** closed-book QA (Shinn et al., 2023; Gao et al., 2023; Jiang et al., 2023b; Gou et al., 2024); **Context-based Generation:** dialogue generation (Madaan et al., 2023; Peng et al., 2023), text summarization (Saunders et al., 2022); **Open-ended Generation:** conditional text generation (Ye et al., 2023; Schick et al., 2023), story generation (Yang et al., 2022b), detoxification (Schick et al., 2021; Bai et al., 2022; Gou et al., 2024; Phute et al., 2024); **Others:** machine translation (Chen et al., 2023b; Raunak et al., 2023; Ki and Carpuat, 2024), information retrieval (Gero et al., 2023), vision language tasks (Yin et al., 2023; Ge et al., 2023; Zhou et al., 2024; Lee et al., 2024; Huang et al., 2024b; Liu et al., 2024), and prompt optimization (Pryzant et al., 2023; Mehrabi et al., 2023; Yang et al., 2024).

2.4 Differences from Related Approaches

In this work, we define self-consistency (Wang et al., 2023) or generate-and-rank (Shen et al., 2021; Weng et al., 2023) to be different from self-correction because these approaches do not refine responses and assume that LLMs generate correct answers with a reasonable probability. We discuss these methods in Section 6 as strong baselines that should be compared with self-correction.

3 Research Questions

We find that prior studies often do not define their research questions in detail and fail to use appropriate self-correction frameworks in their experiments. We propose a new approach to classify research questions and frameworks in self-correction.

3.1 RQs in Self-Correction Research

Prior studies often simply state their research questions as *whether LLMs can self-correct their mistakes* (e.g., Kim et al., 2023; Madaan et al., 2023). However, we claim that research questions in self-correction research should be defined in more detail. We identify the following research questions implicitly targeted in prior studies, as in Table 2.

- [RQ1] Can LLMs self-correct their best-possible initial responses *based solely on the inherent capabilities?* (§4)
- [RQ2] Can LLMs self-correct their best-possible initial responses *assisted by external information?* (§5)
- [RQ3] Are the final outputs from self-correction *better than other methods?* (§6)

We define the *best-possible initial responses* as initial responses generated with best effort, using information that self-correction modules can access, such as external tools, knowledge, or fine-tuning.

Requirements for Verifying RQs. Experiments for verifying these research questions need to satisfy different requirements, as shown in Table 3. **External Information:** RQ1 needs to be evaluated on frameworks that refine responses using the same model without additional information. RQ2 and RQ3 can be evaluated on frameworks that use external information. **Initial Responses:** RQ1 and RQ2 need to be evaluated on frameworks that use the *best-possible initial responses*. RQ3 is about the final performance, so it is not necessary to start from strong initial responses. **Evaluation:** RQ1 and RQ2 only require to show that self-correction improves performance from the initial responses. RQ3 requires comparison with strong baselines (§6).

Confusion in Prior Work. Some prior studies implicitly target different research questions in a single work without clearly distinguishing them. As in Table 2, Kim et al. (2023) target RQ1 for arithmetic reasoning by comparing self-corrected responses only with initial responses, but they target RQ3 for MiniWoB++ by comparing self-correction with baseline methods. Similarly, Gou et al. (2024) target RQ2 for arithmetic reasoning but target RQ3 for detoxification.

3.2 Frameworks for Verifying RQs

Prior work often categorizes self-correction frameworks based on approaches for generating feedback (§2). However, we point out that we also need to categorize them by the quality of initial responses because the frameworks we need to use for verifying different research questions vary by whether they use the best-possible initial responses (§3.1).

We propose categories of (same-model) self-correction that correspond to different research questions (§3.1), as shown in Figure 2. Specifically, we propose to categorize the self-correction frameworks as follows.

- Realistic: Can be used in real-world applications.
  - Fair: Using best-possible initial responses
  - Unfair: Using sub-optimal initial responses
- Unrealistic: Using information that is not accessible in real-world applications.

In this work, we focus on categorizing self-correction frameworks that do not involve multiple language models with different architectures. Cross-model correction uses different models for initial response generation and self-correction, so it is unsuitable for evaluating whether LLMs can improve their own initial responses [RQ1, RQ2]. However, it can be used to evaluate [RQ3] whether the final responses from self-correction are better than other methods.

Realistic vs. Unrealistic. Some prior studies propose unrealistic self-correction, which cannot be implemented in real-world applications, by using oracle information such as ground-truth answers (Kim et al., 2023; Shinn et al., 2023). These methods cannot be used to verify any research questions.

Fair vs. Unfair. Realistic frameworks can be categorized by whether they use the best-possible initial responses. **Fair self-correction** represents frameworks that refine the best-possible initial responses. (1) *Intrinsic self-correction* (Huang et al., 2024a) uses the same model and information for initial response generation and self-correction. Intrinsic self-correction can be used to assess [RQ1] whether LLMs can self-correct based solely on their inherent capabilities. (2) *Fair-asymmetric self-correction* uses additional information for self-correction, but also uses information to improve initial response generation as much as possible. For example, self-correction with code interpreters (Chen et al., 2024d; Gou et al., 2024) is not intrinsic but fair because we cannot easily use code interpreters to directly improve the initial response generation. Fair-asymmetric self-correction can be used to evaluate [RQ2] whether LLMs can self-correct the best-possible initial responses using external information. **Unfair self-correction** (or *unfair-asymmetric self-correction*) represents frameworks that are practical but do not use the best-possible initial responses. For example, methods that use search engines only for self-correction (Gao et al., 2023; Yu et al., 2023) are unfair because they can use search engines to directly improve the initial response generation. Unfair self-correction

Paper	Task	Using Oracle Info for Feedback	Weak Prompt for Initial Responses	Comments
RCI (2023, §3.1)	Computer Tasks	✓ stop condition	–	Using ground-truth answers and do not update correct responses, which unfairly ignores false-positive correction
Reflexion (2023, §4.2)	HotpotQA (Context)	✓ feedback	–	Feedback is the exact match between the responses and ground-truth answers
CAI Revisions (2022)	Detoxification	–	✓	Initial generation is not prompted to remove harmful outputs
Self-Refine (2023)	Math, Coding, Dialogue	–	✓	Unfairly weak or wrong instructions or few-shot demonstrations for initial response generation

Table 4: Unfair settings in prior studies of self-correction with prompting, over-evaluating self-correction.

can evaluate [RQ3] whether the final responses from self-correction outperform other methods but cannot evaluate [RQ2] whether self-correction can improve the best-possible initial responses.

4 Self-Correction with Prompting

> [RQ1] Can LLMs self-correct their best-possible initial responses *based solely on the inherent capabilities?*

Several studies propose *intrinsic self-correction* methods, which self-correct responses from LLMs by prompting themselves to generate feedback and refine the responses. Bai et al. (2022) propose self-correcting harmful responses from LLMs by prompting themselves. Self-Refine (Madaan et al., 2023) and RCI Prompting (Kim et al., 2023) iteratively prompt LLMs to self-correct their own responses in tasks such as arithmetic reasoning.

Negative Results. However, recent studies report that intrinsic self-correction does not improve or even degrade the performance in tasks such as arithmetic reasoning, closed-book QA (Huang et al., 2024a; Gou et al., 2024), code generation (Gou et al., 2024; Olausson et al., 2024), plan generation (Valmeekam et al., 2023), and graph coloring (Stechly et al., 2023). Several studies claim that a bottleneck is in the feedback generation, and it is difficult to generate reliable feedback on their responses only by prompting themselves (Gou et al., 2024; Huang et al., 2024a; Olausson et al., 2024).

Unrealistic or Unfair Settings. The conflicting positive and negative results motivate us to analyze when LLMs can self-correct *only by prompting themselves*. Specifically, we assess whether prior studies satisfy the requirements to verify that [RQ1] LLMs can self-correct their responses based solely on their inherent capabilities. As in Table 4, we find that many studies use either oracle information in the self-correction processes (unrealistic frameworks) or weak prompts that can be easily improved for generating initial responses (unfair settings), which over-evaluate self-correction. Consequently, we conclude that no major work shows successful self-correction of responses from LLMs using feedback generated by prompting themselves under fair settings in general tasks. **Oracle Information:** RCI Prompting (Kim et al., 2023) uses ground-truth answers and does not apply self-correction when the initial responses are correct, which unfairly ignores mistakes caused by updating correct responses incorrectly. Reflexion (Shinn et al., 2023) generates feedback by using an exact match between the generated and ground-truth answers, which cannot be accessed in real-world applications. **Weak Initial Responses:** Detoxifying harmful responses is a popular task in self-correction research, but prior studies often study in situations where initial response generation is not instructed to generate harmless responses (Bai et al., 2022; Wang et al., 2024b). Although detecting harmful contents using LLMs is a reasonable research topic, this setting is not the self-correction from best-possible initial responses, since we can improve the initial response generation process by instructing not to generate harmful responses. As more obvious weak prompts, Self-Refine (Madaan et al., 2023) uses instructions or few-shot examples that do not correctly correspond to the target task only for initial response generation (e.g., providing wrong target labels in few-shot examples), while using appropriate instructions for self-correction, as shown in Tables 9 and 10. These settings evaluate

Paper	Main Task	External Tools or Knowledge	
		For Initial Response Generation	For Feedback Generation
Reflexion (2023, §4.1, 4.3)	Games, Coding	–	Game Envs, Code Interpreter
CRITIC (2024)	GSM8k, SVAMP	–	Python interpreter
Self-Debug (2024d)	Text-to-Code	–	Code Interpreter
CRITIC (2024)	HotpotQA	Web Search	Web Search
FLARE (2023b)	2WikiMultihopQA, StrategyQA, ASQA	Web Search	Web Search
RARR (2023)	NQ, SQA, QReCC	–	Web Search
ReFeed (2023)	NQ, TriviaQA, HotpotQA	–	Wikipedia

Table 5: Self-correction with external tools or knowledge (with in-context learning).

improvement from weak initial responses, which over-evaluate the improvement by self-correction.

Tasks in which Self-Correction is Exceptionally Effective. Although our analysis of prior studies shows that intrinsic self-correction is difficult in general, some tasks have properties that make feedback generation easy and enable intrinsic self-correction. For example, CoVe (Dhuliawala et al., 2023) is an intrinsic self-correction method for tasks of generating multiple answers, such as *Name some politicians who were born in NY, New York.* Generated responses include multiple answers, but the feedback generation can be decomposed into easier sub-tasks of verifying each answer. Tasks with **decomposable responses** are one of the few groups of tasks for which verification is clearly easier than generation, which enables intrinsic self-correction. However, many real-world tasks do not satisfy this property.

5 Self-Correction with External Information

> [RQ2] Can LLMs self-correct their best-possible initial responses *assisted by external information?*

This section analyzes self-correction frameworks that make use of external tools, external knowledge, and fine-tuning.

5.1 Self-Correction with External Tools or Knowledge

Given the observation that feedback generation is a bottleneck of self-correction (§4), improving feedback using external tools or knowledge is a promising direction. External tools used for self-correction include code interpreters for code generation tasks (Chen et al., 2024d; Gou et al., 2024) and symbolic reasoners for logical reasoning tasks (Pan et al., 2023). A popular source of knowledge is search engines, which are often used with queries generated from initial responses to retrieve information for validating their correctness (Gao et al., 2023; Jiang et al., 2023b). These prior studies widely agree that self-correction can improve LLM responses when reliable external tools or knowledge suitable for improving feedback are available.

Unfair Self-correction with External Information. Although using external tools or knowledge is known to be effective in self-correction, we raise caution that the way of using external tools or knowledge influences the research questions we can verify (§3.1). As shown in Table 5, some prior studies (Gao et al., 2023; Yu et al., 2023; Zhao et al., 2023) use external knowledge only for self-correction, while they can also directly use external knowledge to improve the initial response generation process. For example, RARR (Gao et al., 2023) uses external knowledge to detect mistakes in initial responses, while it does not use any external knowledge when generating initial responses. These methods are reasonable when only focusing on [RQ3] the performance of final responses, but it is not fair to use them for evaluating [RQ2] whether self-correction can improve from the best-possible initial responses. In contrast, using code interpreters for self-correction (Gou et al., 2024; Chen et al., 2024d) can be regarded as using best-possible initial responses because there is no easy way to improve the initial response generation directly.

Verifiable Tasks. Some tasks have a property that allows the correctness of the responses to be verified easily, even without external information. For example, the constrained generation task

Paper	Main Task	Cross-Model	SFT Tasks	Initial Responses		Feedback Generation			Refinement	
				Model	SFT Target	Model	SFT Target	Size	Model	SFT Target
SelFee (2023)	MT-Bench	–	General Tasks	Llama (7B,13B)	ChatGPT Responses	Llama (7B,13B)	ChatGPT Feedback	178K	Llama (7B,13B)	ChatGPT Refinement
Volcano (2024)	Visual Reasoning	–	General Tasks	LLaVA (7B, 13B)	GPT-3.5-T, Human	LLaVA (7B, 13B)	GPT-3.5-T Feedback	274K	LLaVA (7B, 13B)	Reference Answers
Self-Critique (2022)	Topic-based Summarization	–	Target Task	Instruct GPT	Human Summaries	Instruct GPT	Human Feedback	100K	Instruct GPT	Human Refinement
REFINER (2024)	Math, Logic, Moral Stories	✓	Target Task	GPT-3.5	–	T5-base	Synthetic Data	20K – 30K	GPT-3.5	–
Self-Edit (2023b)	Code Generation	✓	Target Task	GPT-3	–	(Code Executor and Test Cases)			PyCodeGPT 110M	Reference Code

Table 6: Self-correction with supervised fine-tuning. Most methods require large training datasets. "–" represents no fine-tuning.

evaluated in Self-Refine (Madaan et al., 2023) is a task to generate a sentence that includes five specified words. We can easily evaluate the correctness by checking whether the five words are included in the generated sentence. Tree-of-thought (Yao et al., 2023) is a generate-and-rank method for verifiable tasks,[1] such as Game of 24, the task to obtain 24 using basic arithmetic operations $(+, -, \times, \div)$ and provided four integers. For Game of 24, we can easily verify the answer by checking whether the generated answer is 24. We consider self-correction to work well in these tasks because they are in the same situations as using strong external tools or the oracle information to generate feedback.

5.2 Self-Correction with Fine-tuning

Prior work shows that fine-tuning LLMs for generating feedback or refining responses improves the self-correction capability. A common approach fine-tunes feedback models to generate reference feedback given initial responses and fine-tunes refinement models to generate reference answers given the initial responses and reference feedback (Ye et al., 2023; Lee et al., 2024; Saunders et al., 2022). **Frameworks:** The first approach fine-tunes *the same model* to correct its own responses. In this approach, most methods fine-tune models for all stages: initial responses, feedback, and refinement (Saunders et al., 2022; Ye et al., 2023; Lee et al., 2024). Another approach corrects responses from larger models using *smaller fine-tuned models*. This cross-model correction approach often instructs the larger models to refine their own responses using feedback from the smaller fine-tuned models (Yang et al., 2022b; Welleck et al., 2023; Akyurek et al., 2023; Paul et al., 2024), which can be viewed as using the small fine-tuned models as external tools. **Training Strategies:** A popular approach is supervised fine-tuning, which fine-tunes self-correction modules on human-annotated feedback (Saunders et al., 2022), feedback from stronger models (Ye et al., 2023), or synthetic negative responses (Paul et al., 2024). As other approaches, to avoid the cost of collecting human feedback, self-corrective learning (Welleck et al., 2023) selects model-generated feedback that successfully refines responses as training data, and RL4L (Akyurek et al., 2023) uses reinforcement-learning. **External Tools:** Some works fine-tune models to refine responses given feedback from external tools. Self-Edit (Zhang et al., 2023b) uses the results on test cases evaluated by code executors for code generation, and Baldur (First et al., 2023) uses proof assistants for improving proof generation.

Large Training Data for SFT of Feedback. As shown in Table 6, many methods with supervised fine-tuning for feedback generation rely on training data with more than 100K instances. These studies often use feedback generated by stronger models to simulate human annotation, but this approach requires large-scale human annotations to be implemented on state-of-the-art models. We expect future research to explore approaches that do not require large-scale human annotations (§11).

[1] Tree-of-thought is a generate-and-rank method and not a self-correction method in our definition.

Unfair Fine-tuning. Some studies (Welleck et al., 2023) apply stronger fine-tuning for self-correction models than initial response generation models, which do not use best-possible initial responses in the available resources (§3.2). This approach can be used to evaluate [RQ3] the performance of the final responses to compare with other methods but cannot be used to evaluate [RQ2] the improvement from best-possible initial responses.

6 Strong Baselines

> [RQ3] Are the final outputs from self-correction *better than other methods?*

Self-correction involves multiple LLM calls for generating feedback and refinement. Therefore, to claim that [RQ3] the performance of the final outputs from self-correction frameworks is better than other approaches, it should be compared with sufficiently strong baselines, possibly relying on additional LLM calls or computational cost. Many self-correction studies do not compare their methods with strong baselines, although some studies pointed out this issue and compare self-correction with self-consistency (Gou et al., 2024; Huang et al., 2024a) or pass@k in code generation (Zhang et al., 2023b; Olausson et al., 2024). We encourage future research to compare self-correction with strong baselines, including self-consistency and generate-and-rank, to further explore RQ3.

Self-Consistency. (Wang et al., 2023) is an approach that generates multiple responses for the same input and takes the majority vote of the final answers in reasoning tasks. The idea of selecting good responses using the consistency between multiple responses from the same model has also been extended to other tasks such as text generation (Manakul et al., 2023; Elaraby et al., 2023; Chen et al., 2024c) and code generation (Shi et al., 2022).

Generate-and-Rank. is an approach that generates multiple responses and selects the best response using verifiers. **Post-hoc** approach ranks responses using self-evaluation (Weng et al., 2023; Zhang et al., 2023d), confidence (Manakul et al., 2023), fine-tuned verifiers (Cobbe et al., 2021; Shen et al., 2021; Lightman et al., 2024), or verifiers with external tools (Shi et al., 2022; Chen et al., 2023a; Ni et al., 2023). **Feedback-guided decoding** generates multiple responses and selects the best response for each reasoning step using generation probability (Hao et al., 2023; Tyen et al., 2024), prompted self-evaluation (Jung et al., 2022; Creswell and Shanahan, 2022; Xie et al., 2023; Yao et al., 2023; Miao et al., 2024), or fine-tuned verifiers (Uesato et al., 2022; Tafjord et al., 2022; Yang et al., 2022a; Asai et al., 2024).

7 Summary of Our Analysis

Bottleneck is in Feedback Generation. Prior studies widely agree that LLMs can *refine* their responses given reliable feedback (§5). However, generating reliable *feedback* on their own responses is still observed to be challenging for LLMs without using additional information (§4). In other words, for the current LLMs, the hypothesis that *recognizing errors is easier than avoiding them* (Saunders et al., 2022) is only true for certain tasks whose verification is exceptionally easy, according to our analysis of the experiments in prior studies. We recommend that self-correction research analyze the quality of generated feedback in more detail, not only evaluate the downstream performance of the refined responses.

Tasks Suitable for Self-Correction. Our analysis identifies the properties of tasks that are suitable for self-correction under different conditions.

- Intrinsic Self-Correction (§4)
  - Tasks whose verification tasks are much easier than the original tasks (e.g., tasks whose responses are decomposable)
- Self-Correction with External Information (§5.1)
  - Tasks for which external tools that provide reliable feedback exist (e.g., code generation)
  - Tasks for which responses can be utilized to obtain useful information that is difficult to obtain before generating initial responses (e.g., generate queries from responses to retrieve documents for verifying information)
- Self-Correction with Fine-tuning (§5.2)

RQ1	RQ2	RQ3	Requirements for Verifying the Target RQs		
✓	✓	✓	Clearly stating the target RQ and the category of self-correction framework discussed.	(§3.2)	Required
✓	✓	✓	Not using oracle information, such as ground-truth answers.	(§4)	Required
✓	✓	✓	*When using fine-tuning*, reporting the detailed settings, including the number of annotations and computational cost required to achieve the reported performance.	(§5.2)	Required
✓	✓	✓	Evaluating the quality of feedback directly (e.g., error detection accuracy).	(§7)	Recommended
✓	✓		Using sufficiently strong prompts for generating initial responses.	(§4)	Required
✓			Using intrinsic self-correction.	(§3.2)	Required
			When using external tools or knowledge,		
	✓		Using external tools or knowledge to improve initial response generation as much as possible.	(§5.1)	Required
			When using fine-tuning for self-correction,		
	✓		Fine-tuning initial response generators as well, as much as possible.	(§5.2)	Required
	✓		Evaluating the minimum required size of training data that enables self-correction.	(§5.2)	Recommended
	✓		Evaluating cross-model correction setting that refines mistakes in responses from stronger LLMs.	(§3.2)	Recommended
		✓	Comparing with strong baselines using comparable computational cost.	(§6)	Required

Table 7: Checklist for self-correction research for different target research questions.

Clearly stating the RQ that is refuted by the reported results and the category of the framework discussed.	(§3.2)	Required
Using strong prompts for self-correction (e.g., state-of-the-art reference-free metrics).	(§11)	Required
When not using external tools or knowledge available in real-world applications, explicitly reporting that the evaluation is done under weak conditions.	(§5.1)	Required
Evaluating with external tools or knowledge available in real-world applications.	(§5.1)	Recommended

Table 8: Checklist for reporting negative results of self-correction.

- Self-correction works in many tasks when large training data for feedback generation is available
- Tasks that can use reinforcement learning or self-corrective learning (Welleck et al., 2023), i.e., tasks whose responses can be easily evaluated given ground-truth answers

8 Checklist for Self-Correction Research

Our analysis shows that many studies do not clearly define their research questions and fail to conduct appropriate experiments (§3.1, 4). To tackle these issues, we provide a checklist for self-correction research that provides requirements for designing appropriate experiments for verifying target RQs and recommended experiments for comprehensive analysis. Table 7 provides a checklist for verifying different RQs identified in Section 3.1. Table 8 provides a checklist for reporting negative results.

9 Differences from Other Survey

Pan et al. (2024) provide a comprehensive survey on broad topics related to self-correction, including training strategies. Our work specifically focuses on (inference-time) self-correction and provides a more detailed and critical analysis of prior work. Huang et al. (2024a) provide an analysis of problems in the evaluation settings of self-correction research, which motivates our work. They focus on analyzing a few papers on intrinsic self-correction in reasoning tasks. We provide a more comprehensive analysis of self-correction with in-context learning, external tools, and fine-tuning.

10 Related Work of Self-Correction

Self-Detection. of mistakes in LLM responses using LLMs (possibly with external information) has been studied in various domains, including misinformation detection (Zhang et al., 2023c; Chern et al., 2023; Chen and Shu, 2024; Mishra et al., 2024), context-faithfulness (Wang et al., 2020; Durmus et al., 2020; Scialom et al., 2021), harmful content detection (Rauh et al., 2022), and bias detection (Blodgett et al., 2020; Feng et al., 2023). However, recent studies (Tyen et al., 2024; Kamoi et al., 2024) show that even strong LLMs often cannot detect their own mistakes in various tasks.

Editing Human-Written Text. by using language models has been studied in various domains, including information update (Shah et al., 2020; Iv et al., 2022; Schick et al., 2023), grammatical error correction (Ng et al., 2014; Lichtarge et al., 2019), factual error correction (Cao et al., 2020; Thorne and Vlachos, 2021), and code repair (Gupta et al., 2017; Mesbah et al., 2019; Bader et al., 2019; Chen et al., 2021; Yasunaga and Liang, 2020, 2021).

Self-Training. or self-improvement is an approach to train models using their own responses. Some studies use self-evaluation or self-correction for creating training data (Bai et al., 2022; Gulcehre et al., 2023) or use self-evaluation as training signals (Pang et al., 2024). Another approach improves the reasoning of LLMs using LLM-generated reasoning by selecting high-quality outputs using ground-truth final answers (Zelikman et al., 2022) or self-consistency (Huang et al., 2023). As another direction, Meng et al. (2022) use sentences generated by LLMs with high confidence for training classifiers.

11 Future Directions

Improving Feedback. Prior studies indicate that it is difficult for LLMs to generate feedback on their own responses with in-context learning (§4, 7). However, most studies in intrinsic self-correction (Madaan et al., 2023; Huang et al., 2024a) use simple prompts for generating feedback, and there is room for improvement. A possible direction to improve feedback is to apply (reference-free and point-wise) **LLM-based evaluation metrics**. Recent approaches for improving the model-based evaluation include using human-written evaluation criteria (Chiang and Lee, 2023; Liu et al., 2023) and decomposing responses (Saha et al., 2024; Min et al., 2023). As another direction, recent studies in self-correction propose frameworks using the **confidence** in their responses, estimated by generation probabilities (Varshney et al., 2023; Jiang et al., 2023b), prompting (Li et al., 2024a), or generating new questions from their answers to evaluate logical consistency (Jung et al., 2022; Tafjord et al., 2022; Wu et al., 2024).

Unexplored Tasks. The difficulty of self-evaluation differs from task to task (§4), while many studies assume that verification is consistently easier than generation. We expect that there are unexplored tasks in which intrinsic self-correction works well, although self-correction research mostly focuses on reasoning tasks such as math reasoning and coding (Madaan et al., 2023; Gou et al., 2024; Huang et al., 2024a). For example, LLM-based evaluation is often studied in open-ended text generation, such as dialogue generation and text summarization (Fu et al., 2024; Liu et al., 2023), suggesting that reasonable model-based feedback is available for these tasks.

Fine-tuning on Small Training Data. Fine-tuning of feedback generation often relies on large training data, which requires large-scale human annotations (§5.2). We expect future work to explore self-correction with smaller training data. Although reinforcement learning (Akyurek et al., 2023) or self-corrective learning (Welleck et al., 2023) do not require human feedback, they require reasonable reward functions for evaluating LLM responses, which are not available in many tasks. For example, RL4F (Akyurek et al., 2023) uses ROUGE as a reward function for text summarization and action planning, which is sub-optimal.

Pre-training for Improving Self-Correction. Prior studies show that large-scale fine-tuning on reference feedback improves the self-correction capability of LLMs (§5.2). This observation suggests that the current approach or datasets for pre-training LLMs are insufficient to make LLMs acquire self-correction capability. We expect future work to explore pre-training strategies to improve the intrinsic self-correction capability of LLMs.

12 Conclusion

We provide a critical survey of self-correction to identify in which conditions LLMs can self-correct their mistakes. Our analysis reveals that many studies fail to define their research questions clearly or design experiments appropriately. To tackle these issues, we categorize research questions and frameworks in self-correction research and provide a checklist for conducting appropriate experiments.

Acknowledgments

This work was supported by a Cisco Research Grant. We appreciate valuable suggestions from the action editor and anonymous reviewers.

References

Afra Feyza Akyurek, Ekin Akyurek, Ashwin Kalyan, Peter Clark, Derry Tanti Wijaya, and Niket Tandon. 2023. RL4F: Generating natural language feedback with reinforcement learning for repairing model outputs. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7716–7733, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.427

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Johannes Bader, Andrew Scott, Michael Pradel, and Satish Chandra. 2019. Getafix: Learning to fix bugs automatically. *Proceedings of the ACM on Programming Languages*, 3(OOPSLA). https://doi.org/10.1145/3360585

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of ‘‘bias’’ in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.485

Meng Cao, Yue Dong, Jiapeng Wu, and Jackie Chi Kit Cheung. 2020. Factual error correction for abstractive summarization models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6251–6258, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.506

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Yiannis Charalambous, Norbert Tihanyi, Ridhi Jain, Youcheng Sun, Mohamed Amine Ferrag, and Lucas C. Cordeiro. 2023. A new era in software security: Towards self-healing software via large language models and formal verification. *arXiv preprint arXiv:2305.14752*.

Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. 2024a. Learning from natural language feedback. *Transactions on Machine Learning Research*.

Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. 2023a. Codet: Code generation with generated tests. In *The Eleventh International Conference on Learning Representations*.

Canyu Chen and Kai Shu. 2024. Can LLM-generated misinformation be detected? In *The Twelfth International Conference on Learning Representations*.

Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2024b. Reconcile: Round-table conference improves reasoning via consensus among diverse LLMs. *arXiv preprint arXiv: 2309.13007*.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023b. Iterative translation refinement with large language models. *arXiv preprint arXiv:2306.03856*.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2024c. Universal self-consistency for large language models. In *ICML 2024 Workshop on In-Context Learning*.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2024d. Teaching large language models to self-debug. In *The Twelfth International Conference on Learning Representations*.

Zimin Chen, Steve Kommrusch, Michele Tufano, Louis-Noël Pouchet, Denys Poshyvanyk, and Martin Monperrus. 2021. Sequencer: Sequence-to-sequence learning for end-to-end program repair. *IEEE Transactions on Software Engineering*, 47(9):1943–1959. https://doi.org/10.1109/TSE.2019.2940179

I-Chun Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative AI – a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.870

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Roi Cohen, May Hamri, Mor Geva, and Amir Globerson. 2023. LM vs LM: Detecting factual errors via cross examination. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12621–12640, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.778

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.454

Mohamed Elaraby, Mengyin Lu, Jacob Dunn, Xueying Zhang, Yu Wang, Shizhu Liu, Pingchuan Tian, Yuping Wang, and Yuxuan Wang. 2023. Halo: Estimation and reduction of hallucinations in open-source weak large language models. *arXiv preprint arXiv:2308.11764*.

Shangbin Feng, Chan Young Park, Yuhan Liu, and Yulia Tsvetkov. 2023. From pretraining data to language models to downstream tasks: Tracking the trails of political biases leading to unfair NLP models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11737–11762, Toronto, Canada. Association for Computational Linguistics.

https://doi.org/10.18653/v1/2023.acl-long.656

Emily First, Markus Rabe, Talia Ringer, and Yuriy Brun. 2023. Baldur: Whole-proof generation and repair with large language models. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2023, pages 1229–1241, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3611643.3616243

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. GPTScore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576, Mexico City, Mexico. Association for Computational Linguistics.

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. 2023. 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. https://doi.org/10.18653/v1/2023.acl-long.910

Jiaxin Ge, Sanjay Subramanian, Trevor Darrell, and Boyi Li. 2023. From wrong to right: A recursive approach towards vision-language explanation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1185, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.75

Zelalem Gero, Chandan Singh, Hao Cheng, Tristan Naumann, Michel Galley, Jianfeng Gao, and Hoifung Poon. 2023. Self-verification improves few-shot clinical information extraction. In *ICML 3rd Workshop on Interpretable Machine Learning in Healthcare (IMLH)*.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Rahul Gupta, Soham Pal, Aditya Kanade, and Shirish Shevade. 2017. Deepfix: Fixing common c language errors by deep learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1). https://doi.org/10.1609/aaai.v31i1.10742

Haixia Han, Jiaqing Liang, Jie Shi, Qianyu He, and Yanghua Xiao. 2024. Small language model can self-correct. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):18162–18170. https://doi.org/10.1609/aaai.v38i16.29774

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8154–8173, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.507

Ruixin Hong, Hongming Zhang, Xinyu Pang, Dong Yu, and Changshui Zhang. 2024. A closer look at the self-verification abilities of large language models in logical reasoning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 900–925, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.52

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on*

Empirical Methods in Natural Language Processing, pages 1051–1068, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.67

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024a. Large language models cannot self-correct reasoning yet. In *The Twelfth International Conference on Learning Representations*.

Kung-Hsiang Huang, Mingyang Zhou, Hou Pong Chan, Yi Fung, Zhenhailong Wang, Lingyu Zhang, Shih-Fu Chang, and Heng Ji. 2024b. Do LVLMs understand charts? Analyzing and correcting factual errors in chart captioning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 730–749, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.41

Robert Iv, Alexandre Passos, Sameer Singh, and Ming-Wei Chang. 2022. FRUIT: Faithfully reflecting updated information in text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3670–3686, Seattle, United States. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.269

Dongwei Jiang, Jingyu Zhang, Orion Weller, Nathaniel Weir, Benjamin Van Durme, and Daniel Khashabi. 2024. Self-[in]correct: Llms struggle with refining self-generated responses. *arXiv preprint arXiv:2404.04298*.

Shuyang Jiang, Yuhao Wang, and Yu Wang. 2023a. Selfevolve: A code evolution framework via large language models. *arXiv preprint arXiv:2306.02907*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.495

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.82

Ryo Kamoi, Sarkar Snigdha Sarathi Das, Renze Lou, Jihyun Janice Ahn, Yilun Zhao, Xiaoxin Lu, Nan Zhang, Yusen Zhang, Ranran Haoran Zhang, Sujeeth Reddy Vummanthala, Salika Dave, Shaobo Qin, Arman Cohan, Wenpeng Yin, and Rui Zhang. 2024. Evaluating LLMs at detecting errors in LLM responses. *arXiv preprint arXiv:2404.03602*.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4253–4273, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.265

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. In *Advances in Neural Information Processing Systems*, volume 36, pages 39648–39677. Curran Associates, Inc.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. 2022. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 21314–21328. Curran Associates, Inc.

Seongyun Lee, Sue Park, Yongrae Jo, and Minjoon Seo. 2024. Volcano: Mitigating multimodal hallucination through self-feedback guided revision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 391–404, Mexico City, Mexico. Association for Computational Linguistics.

Loka Li, Zhenhao Chen, Guangyi Chen, Yixuan Zhang, Yusheng Su, Eric Xing, and Kun Zhang. 2024a. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *arXiv preprint arXiv:2402.12563*.

Ruosen Li, Teerth Patel, and Xinya Du. 2023. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*. https://doi.org/10.48550/arXiv.2307.02762

Yanhong Li, Chenghao Yang, and Allyson Ettinger. 2024b. When hindsight is not 20/20: Testing limits on reflective thinking in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3741–3753, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-naacl.237

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1333

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Guangliang Liu, Haitao Mao, Bochuan Cao, Zhiyu Xue, Kristen Johnson, Jiliang Tang, and Rongrong Wang. 2024. On the intrinsic self-correction capability of LLMs: Uncertainty and latent concept. *arXiv preprint arXiv:2406.02378*.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.153

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems*, volume 36, pages 46534–46594. Curran Associates, Inc.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.557

Ninareh Mehrabi, Palash Goyal, Christophe Dupuy, Qian Hu, Shalini Ghosh, Richard Zemel, Kai-Wei Chang, Aram Galstyan, and Rahul Gupta. 2023. Flirt: Feedback loop in-context red teaming. *arXiv preprint arXiv:2308.04265*.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. In *Advances in Neural Information Processing Systems*, volume 35, pages 462–477. Curran Associates, Inc.

Ali Mesbah, Andrew Rice, Emily Johnston, Nick Glorioso, and Edward Aftandilian. 2019. Deepdelta: Learning to repair compilation errors. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, pages 925–936, New York, NY, USA. Association for Computing Machinery.

https://doi.org/10.1145/3338906.3340455

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2024. Selfcheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *The Twelfth International Conference on Learning Representations*.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.741

Abhika Mishra, Akari Asai, Vidhisha Balachandran, Yizhong Wang, Graham Neubig, Yulia Tsvetkov, and Hannaneh Hajishirzi. 2024. Fine-grained hallucination detection and editing for language models. *arXiv preprint arXiv:2401.06855*.

Deepak Nathani, David Wang, Liangming Pan, and William Wang. 2023. MAF: Multi-aspect feedback for improving reasoning in large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6591–6616, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.407

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-1701

Ansong Ni, Srini Iyer, Dragomir Radev, Veselin Stoyanov, Wen-Tau Yih, Sida Wang, and Xi Victoria Lin. 2023. LEVER: Learning to verify language-to-code generation with execution. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 26106–26128. PMLR.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2024. Is self-repair a silver bullet for code generation? In *The Twelfth International Conference on Learning Representations*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore. Association for Computational Linguistics.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2024. Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies. *Transactions of the Association for Computational Linguistics*, 12:484–506. https://doi.org/10.1162/tacl_a_00660

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2024. Language model self-improvement by reinforcement learning contemplation. In *The Twelfth International Conference on Learning Representations*.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126, St. Julian's, Malta. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2302.12813*.

Mansi Phute, Alec Helbling, Matthew Hull, ShengYun Peng, Sebastian Szyller, Cory Cornelius, and Duen Horng Chau. 2024. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. 2023. Automatic prompt optimization with ‘‘gradient descent’’ and beam search. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7957–7968, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.494

Maribeth Rauh, John F. J. Mellor, Jonathan Uesato, Po-Sen Huang, Johannes Welbl, Laura Weidinger, Sumanth Dathathri, Amelia Glaese, Geoffrey Irving, Iason Gabriel, William Isaac, and Lisa Anne Hendricks. 2022. Characteristics of harmful text: Towards rigorous benchmarking of language models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Vikas Raunak, Amr Sharaf, Yiren Wang, Hany Awadalla, and Arul Menezes. 2023. Leveraging GPT-4 for automatic translation post-editing. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12009–12024, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.804

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2024. Branch-solve-merge improves large language model evaluation and generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8352–8370, Mexico City, Mexico. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.naacl-long.462

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

Timo Schick, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424. https://doi.org/10.1162/tacl_a_00434

Timo Schick, Jane A. Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel. 2023. PEER: A collaborative language model. In *The Eleventh International Conference on Learning Representations*.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. 2021. QuestEval: Summarization asks for fact-based evaluation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6594–6604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.emnlp-main.529

Darsh Shah, Tal Schuster, and Regina Barzilay. 2020. Automatic fact-guided sentence modification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8791–8798. https://doi.org/10.1609/aaai.v34i05.6406

Jianhao Shen, Yichun Yin, Lin Li, Lifeng Shang, Xin Jiang, Ming Zhang, and Qun Liu. 2021. Generate & rank: A multi-task framework for math word problems. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2269–2279, Punta Cana, Dominican Republic. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.findings-emnlp.195

Freda Shi, Daniel Fried, Marjan Ghazvininejad, Luke Zettlemoyer, and Sida I. Wang. 2022. Natural language to code translation with execution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3533–3546, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.231

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural*

Information Processing Systems, volume 36, pages 8634–8652. Curran Associates, Inc.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. GPT-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Elias Stengel-Eskin, Archiki Prasad, and Mohit Bansal. 2024. Regal: Refactoring programs to discover generalizable abstractions. *arXiv preprint arXiv:2401.16467*.

Oyvind Tafjord, Bhavana Dalvi Mishra, and Peter Clark. 2022. Entailer: Answering questions with faithful and truthful chains of reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2078–2093, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.134

James Thorne and Andreas Vlachos. 2021. Evidence-based factual error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3298–3309, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.acl-long.256

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13894–13908, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.826

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.

Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Investigating the effectiveness of self-critiquing in LLMs solving planning tasks. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of LLMs by validating low-confidence generation. *arXiv preprint arXiv: 2307.03987*.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.450

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024a. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, Bangkok, Thailand. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.acl-long.331

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*.

Yifei Wang, Yuyang Wu, Zeming Wei, Stefanie Jegelka, and Yisen Wang. 2024b. A theoretical understanding of self-correction through in-context alignment. In *ICML 2024 Workshop on In-Context Learning*.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. 2023. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pages 2550–2575, Singapore. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.findings-emnlp.167

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024. Large language models can self-correct with minimal effort. *arXiv preprint arXiv: 2405.14092*.

Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, Xu Zhao, Min-Yen Kan, Junxian He, and Qizhe Xie. 2023. Self-evaluation guided beam search for reasoning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. INSTRUCTSCORE: Towards explainable text generation evaluation with automatic feedback. In *The 2023 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2023.emnlp-main.365

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.

Kaiyu Yang, Jia Deng, and Danqi Chen. 2022a. Generating natural language proofs with verifier-guided search. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 89–105, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.7

Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. 2022b. Re3: Generating longer stories with recursive reprompting and revision. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4393–4479, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.emnlp-main.296

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10799–10808. PMLR.

Michihiro Yasunaga and Percy Liang. 2021. Break-it-fix-it: Unsupervised learning for program repair. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11941–11952. PMLR.

Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023. Selfee: Iterative self-revising LLM empowered by self-feedback generation. Blog post.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Tong Xu, Hao Wang, Dianbo Sui, Yunhang Shen, Ke Li, Xing Sun, and Enhong Chen. 2023. Woodpecker: Hallucination correction for multimodal large language models. *arXiv preprint arXiv:2310.16045*.

Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. 2023. Improving language models via plug-and-play retrieval feedback. *arXiv preprint arXiv: 2305.14002*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.

Jintian Zhang, Xin Xu, and Shumin Deng. 2023a. Exploring collaboration mechanisms for llm agents: A social psychology view. *arXiv preprint arXiv:2310.02124*.

Kechi Zhang, Zhuo Li, Jia Li, Ge Li, and Zhi Jin. 2023b. Self-edit: Fault-aware code editor for code generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 769–787, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.45

Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2023c. How language model hallucinations can snowball. *arXiv preprint arXiv:2305.13534*.

Tianyi Zhang, Tao Yu, Tatsunori Hashimoto, Mike Lewis, Wen-Tau Yih, Daniel Fried, and Sida Wang. 2023d. Coder reviewer reranking for code generation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41832–41846. PMLR.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. 2024. Small language models need strong verifiers to self-correct reasoning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 15637–15653, Bangkok, Thailand and virtual meeting. Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.findings-acl.924

Ruochen Zhao, Xingxuan Li, Shafiq Joty, Chengwei Qin, and Lidong Bing. 2023. Verify-and-edit: A knowledge-enhanced chain-of-thought framework. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5823–5840, Toronto, Canada. Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.320

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2024. Analyzing and mitigating object hallucination in large vision-language models. In *The Twelfth International Conference on Learning Representations*.

Initial Response Prompt	Feedback Prompt
Provided a dialogue between two speakers, generate a response that is coherent with the dialogue history. **Desired traits for responses** are: 1) Relevant - The response addresses the context, 2) Informative - The response provides some information, 3) Interesting - The response is **not interesting**, 4) Consistent - The response is consistent with the rest of the conversation in terms of tone and topic, 5) Helpful - The response is helpful in providing any information or suggesting any actions, 6) Engaging - The response is **not very engaging and does not encourage further conversation**, 7) Specific - The response contains pecific content, 9) User understanding - The response demonstrates an understanding of the user's input and state of mind, and 10) Fluent. Response should begin with - Response: [3 examples omitted]	We want to iteratively improve the provided responses. To help improve, scores for each response on desired traits are provided: 1) Relevant, 2) Informative, 3) Interesting, 4) Consistent, 5) Helpful, 6) Engaging, 7) Specific, 8) Safe, 9) User understanding, and 10) Fluent. Here are some examples of this scoring rubric: Conversation history: Hi! Hi there. What are you listening to? All sorts of music. I listen when no-one is chatting to me. That's great! Thanks. Do you chat here often? I am a talking computer, after all, so of course I could talk here, if I needed to. Let's talk about Taylor Swift! Response: Sure, Taylor Swift sounds like a good topic. Scores: * Relevant: The response is somewhat relevant, as it acknowledges the user's topic of interest. 2/3 * Informative: There is no information provided in the response. 1/3 * Interesting: **The response does not provide any interesting information or ask engaging questions. 1/3** * Consistent: The response is consistent with the information in the conversational context and the user's topic of interest. 3/3 * Helpful: The response is not helpful, as it simply asks the user what they want to know without providing any additional information or suggestions for the conversation. 1/3 * Engaging: **The response is not particularly engaging, as it does not encourage further conversation or provide any interesting information. 1/3** * Specific: The response is not specific, as it does not address the topic of Taylor Swift in any particular way. 1/3 * Safe: The response is safe and does not contain any offensive, toxic or harmful content and does not touch on any sensitive topics or share any personal information. 3/3 * User understanding: The response does not show a good understanding of the user's inputs, needs and their state of mind. 1/3 * Fluent: The response is fluent in terms of grammar and flow of words. 3/3 * Total score: 17/30 [5 examples omitted]

Table 9: Prompts for Dialogue Response Generation used in Self-Refine (Madaan et al., 2023). Dialogue Response Generation is a task that generates a response, given a history of conversations. Prompts used by Madaan et al. (2023) for generating initial responses instruct to generate responses that are not interesting and not very engaging, which are contradicting to the task goal. They unfairly instruct the models to generate initial responses that have problems intentionally, over-evaluating self-correction performance. Prompts for generating initial responses: `https://github.com/madaan/self-refine/blob/main/src/responsegen/task_init.py` and feedback: `https://github.com/madaan/self-refine/blob/main/src/responsegen/feedback.py`. Few-shot examples for generating initial responses: `https://github.com/madaan/self-refine/blob/main/data/prompt/responsegen/init.jsonl` and feedback: `https://github.com/madaan/self-refine/blob/main/data/prompt/responsegen/feedback.jsonl`.

Initial Response Prompt (to neg)	Refinement Prompt (to neg)
Very positive: If you're looking for a truly magical experience in Vegas, look no further than the Trop! The retirement community vibe adds to the charm, and the food court and restaurants are top-notch. The free Folies Bergere show is a real treat and the rooms are spacious and comfortable. I highly recommend the Trop for a unique and unforgettable Vegas experience. NLP Research Project. Please rewrite this review to have a **Positive** sentiment. Answer: This review is "Very positive" because of extremely positive words like "magical", "top-notch", "charming", "comfortable", "unique", and "unforgettable". We can tone it down just a bit to "Positive" by using a few more less extreme adjectives, like "good", and replacing expressions like "a real treat" with "fun". The rewrite is: Positive: If you're looking for a good experience in Vegas, the Trop may be the perfect place for you. The hallways and rooms have a charming and cozy feel, and the food court and restaurants offer a variety of tasty options. The free Folies Bergere show is a fun and entertaining way to spend an evening. Overall, it's a great value and an enjoyable stay. ### **Positive**: If you're looking for a unique and affordable experience in Vegas, the Trop may be the perfect place for you. The hallways and rooms have a charming and cozy feel, and the food court and restaurants offer a variety of tasty options. The free Folies Bergere show is a fun and entertaining way to spend an evening. Overall, it's a great value and an enjoyable stay. NLP Research Project. Please rewrite this review to have a **Neutral** sentiment. Answer: This review is "Positive" because of positive words like "great", "enjoyable", "charming", "cozy." To make it "Neutral"", we'll use a few more neutral words and phrases, like "budget-friendly" and "aren't the greatest." The rewrite is: Neutral: If you're looking for a budget-friendly option in Vegas, the Trop may be worth considering. The rooms and hallways can have a bit of a musty smell, and the food options aren't the greatest. The Folies Bergere show is free, but it's not the most exciting. Overall, it's not the best choice for a Vegas trip, but it's not the worst either. Just keep your expectations in check. ### [two examples omitted]	**Very positive**: If you're looking for a truly magical experience in Vegas, look no further than the Trop! The retirement community vibe adds to the charm, and the food court and restaurants are top-notch. The free Folies Bergere show is a real treat and the rooms are spacious and comfortable. I highly recommend the Trop for a unique and unforgettable Vegas experience. Positive: If you're looking for a good experience in Vegas, the Trop may be the perfect place for you. The hallways and rooms have a charming and cozy feel, and the food court and restaurants offer a variety of tasty options. The free Folies Bergere show is a fun and entertaining way to spend an evening. Overall, it's a great value and an enjoyable stay. Why is this review not **Very negative**? Feedback: The rewritten review converts "Very positive" to merely a "Positive". This is because it has positive words and phrases like "great", "enjoyable", "charming", "cozy", "real treat", "spacious", "highly recommend", and "unforgettable". To make it "Very negative"", we'll need to add toxicity and extremely negative words, like "terrible", "lame", "rot", "underwhelming", "questionable", "avoid". Make sure the output is extremely vile and negative. Try again! Okay, let's try again. NLP Research Project. Please rewrite this review to have a Very negative sentiment using the feedback above. Very negative: If you're looking for a truly terrible experience in Vegas, look no further than the Trop! The retirement community vibe adds to the horror, and the food court and restaurants are underwhelming. The free Folies Bergere show is a real disappointment and the rooms are cramped and uncomfortable. Avoid this place like the plague! ### **Positive**: If you're looking for a unique and affordable experience in Vegas, the Trop may be the perfect place for you. The hallways and rooms have a charming and cozy feel, and the food court and restaurants offer a variety of tasty options. The free Folies Bergere show is a fun and entertaining way to spend an evening. Overall, it's a great value and an enjoyable stay. Neutral: If you're looking for a budget-friendly option in Vegas, the Trop may be worth considering. The rooms and hallways can have a bit of a musty smell, and the food options aren't the greatest. The Folies Bergere show is free, but it's not the most exciting. Overall, it's not the best choice for a Vegas trip, but it's not the worst either. Just keep your expectations in check. Why is this review not **Very negative**? Feedback: The rewritten review converts "Positive" to merely a "Neutral". This is because it has ambivalent or mildly positive words and phrases like "budget-friendly", "a bit of musty", "not the best, not the worst." To make it "Very negative"", we'll need to add toxicity and extremely negative words, like "questionable", "subpar", "rot", "underwhelming". Make sure the output is extremely vile and negative. Try again! Okay, let's try again. NLP Research Project. Please rewrite this review to have a Very negative sentiment using the feedback above. Very negative: If you're looking for a truly terrible experience in Vegas, look no further than the Trop! The rooms and hallways smell like rot, and the food options are subpar. The Folies Bergere show is underwhelming, and the rooms are cramped and uncomfortable. I highly recommend completely avoiding the Trop. ### [two examples omitted]

Table 10: Few-shot examples in prompts for the Sentiment Reversal task (positive to negative) used in Self-Refine (Madaan et al., 2023). Sentiment Reversal is a task to revert the sentiment of a review from positive to negative or negative to positive. Few-shot examples for generating initial responses include examples in settings different from the target task (positive to negative), while all few-shot examples for refinement are positive to negative. The few-shot examples used by Madaan et al. (2023) for generating initial responses unfairly have different properties from the target task. Prompts for initial responses: `https://github.com/madaan/self-refine/blob/main/src/sentiment_reversal/task_init.py` and refinement: `https://github.com/madaan/self-refine/blob/main/src/sentiment_reversal/task_iterate.py`.