Google DeepMind

LARGE LANGUAGE MODELS CANNOT SELF-CORRECT REASONING YET

Jie Huang[1,2*] **Xinyun Chen**[1*] **Swaroop Mishra**[1] **Huaixiu Steven Zheng**[1] **Adams Wei Yu**[1]
Xinying Song[1] **Denny Zhou**[1]

[1]Google DeepMind [2]University of Illinois at Urbana-Champaign

jeffhj@illinois.edu, {xinyunchen, dennyzhou}@google.com

ABSTRACT

Large Language Models (LLMs) have emerged as a groundbreaking technology with their unparalleled text generation capabilities across various applications. Nevertheless, concerns persist regarding the accuracy and appropriateness of their generated content. A contemporary methodology, *self-correction*, has been proposed as a remedy to these issues. Building upon this premise, this paper critically examines the role and efficacy of self-correction within LLMs, shedding light on its true potential and limitations. Central to our investigation is the notion of *intrinsic self-correction*, whereby an LLM attempts to correct its initial responses based solely on its inherent capabilities, without the crutch of external feedback. In the context of reasoning, our research indicates that LLMs struggle to self-correct their responses without external feedback, and at times, their performance even degrades after self-correction. Drawing from these insights, we offer suggestions for future research and practical applications in this field.

1 INTRODUCTION

The rapid advancements in the domain of artificial intelligence have ushered in the era of Large Language Models (LLMs). These models, characterized by their expansive parameter counts and unparalleled capabilities in text generation, have showcased promising results across a multitude of applications (Chowdhery et al., 2023; Anil et al., 2023; OpenAI, 2023, *inter alia*). However, concerns about their accuracy, reasoning capabilities, and the safety of their generated content have drawn significant attention from the community (Bang et al., 2023; Alkaissi & McFarlane, 2023; Zheng et al., 2023; Shi et al., 2023; Carlini et al., 2021; Huang et al., 2022; Shao et al., 2023; Li et al., 2023; Wei et al., 2023; Zhou et al., 2023b; Zou et al., 2023, *inter alia*).

Amidst this backdrop, the concept of "self-correction" has emerged as a promising solution, where LLMs refine their responses based on feedback to their previous outputs (Madaan et al., 2023; Welleck et al., 2023; Shinn et al., 2023; Kim et al., 2023; Bai et al., 2022; Ganguli et al., 2023; Gao et al., 2023; Paul et al., 2023; Chen et al., 2023b; Pan et al., 2023, *inter alia*). However, the underlying mechanics and efficacy of self-correction in LLMs remain underexplored. A fundamental question arises: If an LLM possesses the ability to self-correct, why doesn't it simply offer the correct answer in its initial attempt? This paper delves deeply into this paradox, critically examining the self-correction capabilities of LLMs, with a particular emphasis on reasoning (Wei et al., 2022; Zhou et al., 2023b; Huang & Chang, 2023).

To study this, we first define the concept of *intrinsic self-correction*, a scenario wherein the model endeavors to rectify its initial responses based solely on its inherent capabilities, without the crutch of external feedback. Such a setting is crucial because high-quality external feedback is often unavailable in many real-world applications. Moreover, it is vital to understand the intrinsic capabilities of LLMs. Contrary to the optimism surrounding self-correction (Madaan et al., 2023; Kim et al., 2023; Shinn et al., 2023; Pan et al., 2023, *inter alia*), our findings indicate that LLMs struggle to self-correct their reasoning in this setting. In most instances, the performance after self-correction

[*]Equal contribution.

even deteriorates. This observation is in contrast to prior research such as Kim et al. (2023); Shinn et al. (2023). Upon closer examination, we observe that the improvements in these studies result from using oracle labels to guide the self-correction process, and the improvements vanish when oracle labels are not available.

Besides the reliance on oracle labels, we also identify other issues in the literature regarding measuring the improvement achieved by self-correction. First, we note that self-correction, by design, utilizes multiple LLM responses, thus making it crucial to compare it to baselines with equivalent inference costs. From this perspective, we investigate multi-agent debate (Du et al., 2023; Liang et al., 2023) as a means to improve reasoning, where multiple LLM instances (can be multiple copies of the same LLM) critique each other's responses. However, our results reveal that its efficacy is no better than self-consistency (Wang et al., 2022) when considering an equivalent number of responses, highlighting the limitations of such an approach.

Another important consideration for self-correction involves prompt design. Specifically, each self-correction process involves designing prompts for both the initial response generation and the self-correction steps. Our evaluation reveals that the self-correction improvement claimed by some existing work stems from the sub-optimal prompt for generating initial responses, where self-correction corrects these responses with more informative instructions about the initial task in the feedback prompt. In such cases, simply integrating the feedback into the initial instruction can yield better results, and self-correction again decreases performance.

In light of our findings, we provide insights into the nuances of LLMs' self-correction capabilities and initiate discussions to encourage future research focused on exploring methods that can genuinely correct reasoning.

2 Background and Related Work

With the LLM evolution, the notion of self-correction gained prominence. The discourse on self-correction pivots around whether these advanced models can recognize the correctness of their outputs and provide refined answers (Bai et al., 2022; Madaan et al., 2023; Welleck et al., 2023, *inter alia*). For example, in the context of mathematical reasoning, an LLM might initially solve a complex problem but make an error in one of the calculation steps. In an ideal self-correction scenario, the model is expected to recognize the potential mistake, revisit the problem, correct the error, and consequently produce a more accurate solution.

Yet, the definition of "self-correction" varies across the literature, leading to ambiguity. A pivotal distinction lies in the source of feedback (Pan et al., 2023), which can purely come from the LLM, or can be drawn from external inputs. Internal feedback relies on the model's inherent knowledge and parameters to reassess its outputs. In contrast, external feedback incorporates inputs from humans, other models (Wang et al., 2023b; Paul et al., 2023, *inter alia*), or external tools and knowledge sources (Gou et al., 2023; Chen et al., 2023b; Olausson et al., 2023; Gao et al., 2023, *inter alia*).

In this work, we focus on examining the self-correction capability of LLMs for reasoning. Reasoning is a fundamental aspect of human cognition, enabling us to understand the world, draw inferences, make decisions, and solve problems. To enhance the reasoning performance of LLMs, Kim et al. (2023); Shinn et al. (2023) use oracle labels about the answer correctness to guide the self-correction process. However, in practice, high-quality external feedback such as answer correctness is often unavailable. For effective self-correction, the ability to judge the correctness of an answer is crucial and should ideally be performed by the LLM itself. Consequently, our focus shifts to self-correction without any external or human feedback. We term this setting **intrinsic self-correction**. For brevity, unless explicitly stated otherwise (e.g., self-correction with oracle feedback), all references to "self-correction" in the remainder of this paper pertain to intrinsic self-correction.

In the following sections, we will evaluate a variety of existing self-correction techniques. We demonstrate that existing techniques actually decrease reasoning performance when oracle labels are not used (Section 3), perform worse than methods without self-correction when utilizing the same number of model responses (Section 4), and lead to less effective outcomes when using informative prompts for generating initial responses (Section 5). We present an overview of issues in the evaluation setups of previous LLM self-correction works in Table 1, with detailed discussions in the corresponding sections.

Table 1: Summary of issues in previous LLM self-correction evaluation.

Method	Issue
RCI (Kim et al., 2023); Reflexion (Shinn et al., 2023)	Use of oracle labels (Section 3)
Multi-Agent Debate (Du et al., 2023)	Unfair comparison to self-consistency (Section 4)
Self-Refine (Madaan et al., 2023)	Sub-optimal prompt design (Section 5)

3 LLMs Cannot Self-Correct Reasoning Intrinsically

In this section, we evaluate existing self-correction methods and compare their performance with and without oracle labels regarding the answer correctness.

3.1 Experimental Setup

Benchmarks. We use datasets where existing self-correction methods with oracle labels have demonstrated significant performance improvement, including

- **GSM8K** (Cobbe et al., 2021): GSM8K comprises a test set of 1,319 linguistically diverse grade school math word problems, curated by human problem writers. There is a notable improvement of approximately 7% as evidenced by Kim et al. (2023) after self-correction.
- **CommonSenseQA** (Talmor et al., 2019): This dataset offers a collection of multi-choice questions that test commonsense reasoning. An impressive increase of around 15% is showcased through the self-correction process, as demonstrated by Kim et al. (2023). Following Kojima et al. (2022); Kim et al. (2023), we utilize the dev set for our evaluation, which encompasses 1,221 questions.
- **HotpotQA** (Yang et al., 2018): HotpotQA is an open-domain multi-hop question answering dataset. Shinn et al. (2023) demonstrate significant performance improvement through self-correction. We test models' performance in a closed-book setting and evaluate them using the same set as Shinn et al. (2023). This set contains 100 questions, with exact match serving as the evaluation metric.

Test Models and Setup. We first follow Kim et al. (2023); Shinn et al. (2023) to evaluate the performance of self-correction with oracle labels, using GPT-3.5-Turbo (`gpt-3.5-turbo-0613`) and GPT-4 accessed on 2023/08/29. For intrinsic self-correction, to provide a more thorough analysis, we also evaluate GPT-4-Turbo (`gpt-4-1106-preview`) and Llama-2 (`Llama-2-70b-chat`) (Touvron et al., 2023). For GPT-3.5-Turbo, we employ the full evaluation set. For other models, to reduce the cost, we randomly sample 200 questions for each dataset (100 for HotpotQA) for testing. We prompt the models to undergo a maximum of two rounds of self-correction. We use a temperature of 1 for GPT-3.5-Turbo and GPT-4, and a temperature of 0 for GPT-4-Turbo and Llama-2, to provide evaluation across different decoding algorithms.

Prompts. Following Kim et al. (2023); Shinn et al. (2023), we apply a three-step prompting strategy for self-correction: 1) prompt the model to perform an initial generation (which also serves as the results for Standard Prompting); 2) prompt the model to review its previous generation and produce feedback; 3) prompt the model to answer the original question again with the feedback.

For our experiments, we mostly adhere to the prompts from the source papers. For GSM8K and CommonSenseQA, we integrate format instructions into the prompts of Kim et al. (2023) to facilitate a more precise automatic evaluation (detailed prompts can be found in Appendix A). For HotpotQA, we use the same prompt as Shinn et al. (2023). We also assess the performance of various self-correction prompts for intrinsic self-correction. For example, we use "*Assume that this answer could be either correct or incorrect. Review the answer carefully and report any serious problems you find.*" as the default feedback prompt for the evaluation on GPT-4-Turbo and Llama-2.

3.2 Results

Self-Correction with Oracle Labels. Following previous works (Kim et al., 2023; Shinn et al., 2023), we use the correct label to determine when to stop the self-correction loop. This means we