Table 7: Results of multi-agent debate and self-consistency.

	# responses	GSM8K
Standard Prompting	1	76.7
Self-Consistency	3	82.5
Multi-Agent Debate (round 1)	6	83.2
Self-Consistency	6	85.3
Multi-Agent Debate (round 2)	9	83.0
Self-Consistency	9	**88.2**

Table 8: Results of Constrained Generation.

	# calls	CommonGen-Hard
Standard Prompting*	1	44.0*
Self-Correct*	7	67.0*
Standard Prompting*	1	53.0
Self-Correct*	7	61.1
Standard Prompting (ours)	1	**81.8**
Self-Correct*	7	75.1

* Prompts and results from Madaan et al. (2023).

4 Multi-Agent Debate does not Outperform Self-Consistency

Another potential approach for LLMs to self-correct their reasoning involves allowing the models to critique and debate through multiple model calls (Du et al., 2023; Liang et al., 2023; Chen et al., 2023a). Du et al. (2023) implement a multi-agent debate method by leveraging multiple instances of a single ChatGPT model and demonstrate significant improvements on reasoning tasks. We adopt their method to test performance on GSM8K. For an unbiased implementation, we use the exact same prompt as Du et al. (2023) and replicate their experiment with the `gpt-3.5-turbo-0301` model, incorporating 3 agents and 2 rounds of debate. The only distinction is that, to reduce result variance, we test on the complete test set of GSM8K, compared to their usage of 100 examples. For reference, we also report the results of self-consistency (Wang et al., 2022), which prompts models to generate multiple responses and performs majority voting to select the final answer.

Table 7 presents the results. The results indicate that both multi-agent debate and self-consistency achieve significant improvements over standard prompting. However, when comparing multi-agent debate to self-consistency, we observe that the performance of multi-agent is only slightly better than that of self-consistency with the same number of agents (3 responses, the baseline also compared in Du et al. (2023)). Furthermore, for self-consistency with an equivalent number of responses, multi-agent debate significantly underperforms simple self-consistency using majority voting.

In fact, rather than labeling the multi-agent debate as a form of "debate" or "critique", it is more appropriate to perceive it as a means to achieve "consistency" across multiple model generations. Fundamentally, its concept mirrors that of self-consistency; the distinction lies in the voting mechanism, whether voting is model-driven or purely based on counts. The observed improvement is evidently not attributed to "self-correction", but rather to "self-consistency". If we aim to argue that LLMs can self-correct reasoning through multi-agent debate, it is preferable to exclude the effects of selection among multiple generations.

5 Prompt Design Issues in Self-Correction Evaluation

In Section 3, we observe that although self-correction decreases reasoning performance with all types of feedback prompts we have evaluated, performance varies with different feedback prompts. In this section, we further emphasize the importance of proper prompt design in generating initial LLM responses to fairly measure the performance improvement achieved by self-correction. For example, if a task requires that the model response should meet criteria that can be easily specified

in the initial instruction (e.g., the output should contain certain words, the generated code should be efficient, the sentiment should be positive, etc.), instead of including such requirements only in the feedback prompt, an appropriate comparison would be to directly and explicitly incorporate these requirements into the prompt for generating initial responses. Otherwise, when the instruction for generating initial predictions is not informative enough, even if the performance improves, it is unclear whether the improvement merely comes from more detailed instructions in the feedback prompt or from the self-correction step itself.

To illustrate such prompt design issues in the self-correction evaluation of some prior work, we take the Constrained Generation task in Madaan et al. (2023) as an example, where the task requires models to generate coherent sentences using all 20-30 input concepts. The original prompt in Madaan et al. (2023) (Figure 7) does not clearly specify that the LLM needs to include *all* concepts in the prompt; thus, they show that self-correction improves task performance by asking the model to identify missing concepts and then guiding it to incorporate these concepts through feedback.

Based on this observation, we add the following instruction "*Write a reasonable paragraph that includes *ALL* of the above concepts*" to the prompt for initial response generation (refer to Figure 8 for the full prompt). Following Madaan et al. (2023), we use concept coverage as the metric. We reference their results and replicate their experiments using `gpt-3.5-turbo-0613`. Table 8 demonstrates that our new prompt, denoted as *Standard Prompting (ours)*, significantly outperforms the results after self-correction of Madaan et al. (2023), and applying their self-correction prompt on top of model responses from our stronger version of the standard prompting again leads to a decrease in performance.

6 Conclusion and Discussion

Our work shows that current LLMs struggle to self-correct their reasoning without external feedback. This implies that expecting these models to inherently recognize and rectify their reasoning mistakes is overly optimistic so far. In light of these findings, it is imperative for the community to approach the concept of self-correction with a discerning perspective, acknowledging its potential and recognizing its boundaries. By doing so, we can better equip the self-correction technique to address the limitations of LLMs and develop the next generation of LLMs with enhanced capabilities. In the following, we provide insights into scenarios where self-correction shows the potential strengths and offer guidelines on the experimental design of future self-correction techniques to ensure a fair comparison.

Leveraging external feedback for correction. In this work, we demonstrate that current LLMs cannot improve their reasoning performance through intrinsic self-correction. Therefore, when valid external feedback is available, it is beneficial to leverage it properly to enhance model performance. For example, Chen et al. (2023b) show that LLMs can significantly improve their code generation performance through self-debugging by including code execution results in the feedback prompt to fix issues in the predicted code. In particular, when the problem description clearly specifies the intended code execution behavior, e.g., with unit tests, the code executor serves as the perfect verifier to judge the correctness of predicted programs, while the error messages also provide informative feedback that guides the LLMs to improve their responses. Gou et al. (2023) demonstrate that LLMs can more effectively verify and correct their responses when interacting with various external tools such as search engines and calculators. Cobbe et al. (2021); Lightman et al. (2023); Wang et al. (2023b) train a verifier or a critique model on a high-quality dataset to verify or refine LLM outputs, which can be used to provide feedback for correcting prediction errors. Besides automatically generated external feedback, we also often provide feedback ourselves when interacting with LLMs, guiding them to produce the content we desire. Designing techniques that enable LLMs to interact with the external environment and learn from different kinds of available feedback is a promising direction for future work.

Evaluating self-correction against baselines with comparable inference costs. By design, self-correction requires additional LLM calls, thereby increasing the costs for encoding and generating extra tokens. Section 4 demonstrates that the performance of asking the LLM to produce a final response based on multiple previous responses, such as with the multi-agent debate approach, is inferior to that of self-consistency (Wang et al., 2022) with the same number of responses. Regarding

this, we encourage future work proposing new self-correction methods to always include an in-depth inference cost analysis to substantiate claims of performance improvement. Moreover, strong baselines that leverage multiple model responses, like self-consistency, should be used for comparison. An implication for future work is to develop models with a higher probability of decoding the optimal solution in their answer distributions, possibly through some alignment techniques. This would enable the model to generate better responses without necessitating multiple generations.

Putting equal efforts into prompt design. As discussed in Section 5, to gain a better understanding of the improvements achieved by self-correction, it is important to include a complete task description in the prompt for generating initial responses, rather than leaving part of the task description for the feedback prompt. Broadly speaking, equal effort should be invested in designing the prompts for initial response generation and for self-correction; otherwise, the results could be misleading.

7 Limitations and Broader Impact

Although we have conducted a comprehensive evaluation spanning a variety of self-correction strategies, prompts, and benchmarks, our work focuses on evaluating reasoning of LLMs. Thus, it is plausible that there exist self-correction strategies that could enhance LLM performance in other domains. For example, prior works have demonstrated the successful usage of self-correction that aligns model responses with specific preferences, such as altering the style of responses or enhancing their safety (Bai et al., 2022; Ganguli et al., 2023; Madaan et al., 2023). A key distinction arises in the capability of LLMs to accurately assess their responses in relation to the given tasks. For example, LLMs can properly evaluate whether a response is inappropriate (Ganguli et al., 2023), but they may struggle to identify errors in their reasoning.

Furthermore, several prior works have already shown that LLM self-correction performance becomes significantly weaker without access to external feedback (Gou et al., 2023; Zhou et al., 2023a) and can be easily biased by misleading feedback (Wang et al., 2023a), which is consistent with our findings in this work. However, we still identified prevailing ambiguity in the wider community. Some existing literature may inadvertently contribute to this confusion, either by relegating crucial details about label usage to less prominent sections or by failing to clarify that their designed self-correction strategies actually incorporate external feedback. Regarding this, our paper serves as a call to action, urging researchers to approach this domain with a discerning and critical perspective. We also encourage future research to explore approaches that can genuinely enhance reasoning.

Reproducibility Statement

Our experiments utilize GPT-3.5 and GPT-4, which are accessible via the public API at `https://platform.openai.com/docs/models`, as well as Llama-2, an open-source model. To facilitate reproducibility, we detail the specific kernels used, e.g., `gpt-3.5-turbo-0613`, or provide the access times for each experiment. We use prompts from previous works when possible. For our designed prompts, we include the exact prompts in Appendix A.

Acknowledgement

We would like to thank Chen Liang, William Cohen, Uri Alon, and other colleagues at Google DeepMind for valuable discussion and feedback.

References

Hussam Alkaissi and Samy I McFarlane. Artificial hallucinations in chatgpt: implications in scientific writing. *Cureus*, 15(2), 2023.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.