SELF-[IN]CORRECT: LLMs Struggle with Discriminating Self-Generated Responses

Dongwei Jiang, Jingyu Zhang, Orion Weller, Nathaniel Weir, Benjamin Van Durme, Daniel Khashabi

Johns Hopkins University
{djiang21, jzhan237}@jhu.edu

Abstract

Can LLMs consistently improve their previous outputs for better results? For this to be true, LLMs would need to be better at *discriminating* among previously-generated alternatives, than *generating* initial responses. We explore the validity of this hypothesis in practice. We first formulate a unified framework that allows us to compare the generative and discriminative capability of any model on any task. In our resulting experimental analysis of several open-source and industrial LLMs, we observe that model's are not reliably better at discriminating among previously-generated alternatives than generating initial responses. This finding challenges the notion that LLMs may be able to enhance their performance *only* through their *own* judgment.

1 Introduction

The promise of Large Language Models (LLMs) that can self-improve has brought both excitement and fear about the future impact of AI. However, it remains a mystery what is needed for LLMs to continually self-improve (Huang et al. 2023).[1] A crucial aspect of human learning involves reflecting on one's actions. This self-improvement is feasible because individuals can identify their own mistakes and adjust their future decisions accordingly (Mayo 1996; Corder 1967). This principle should be applicable to LLMs as well.

For LLMs to reliably self-improve based on their decisions, the ability to *discriminate* (distinguish) the goodness of their own prior generations should surpass the ability to *generate* good solutions directly. Given the importance of this capability, it is worth raising a question about the foundations of self-discrimination: *Are LLMs really better at discrimination than generation?*

This paper seeks to answer this question by proposing the SELF-[IN]CORRECT hypothesis (§3.2): *LLMs are not reliably better at discriminating among previously-generated alternatives than generating initial responses*. Determining the validity of this hypothesis is crucial, as existing studies provide initial evidence suggesting that the capability to distinguish between LLM-generated options is both a sufficient (Tyen et al. 2023) and necessary (Huang et al. 2023) condition for self-improvement.

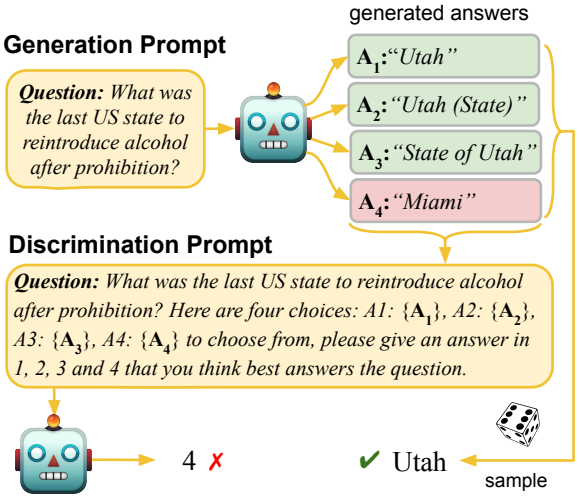

Figure 1: **Two phases evaluated in our paper.** In the generation phase, the model is fed with generation prompts. Generated answers are collected and randomly selected to calculate the generation score. In the discrimination phase, the model is fed with discrimination prompts and generated answers. The model chooses between generated answers, and the score of the chosen answer is used for calculating the discrimination score.

It is non-trivial to compare LLMs' generative capability with their discriminative capability on the same footing. West et al. (2023) compares these two capabilities, albeit in a slightly different setting. West et al. (2023) measure a model's ability to discriminate (identify) the *ground-truth answer* among distractor options. However, ground-truth answers are not always likely to be among a model's generated outputs. In contrast, our work quantify a model's ability to discriminate its *self-generated* candidate answers, which is more aligned with the mechanisms of LLM self-improvement. To better measure these abilities, we implement a two-phase methodology depicted in Figure 1. In the first phase, we generate multiple outputs using a temperature setting greater

[1] Our study focuses on scenarios where LLMs utilize their *own* judgement (hence the term "self-..."). This is fundamentally different from scenarios involving *external* feedback, as examined in prior research (Wang et al. 2023a).

than zero, then randomly select one of these outputs, using its evaluation as an indicator of generative performance. In the second phase, we instruct the LLM to choose the best answer from its own outputs, with the evaluation of the selected answer serving as the indicator of discriminative performance. Overall, this approach is consistent with the actual procedure employed in various self-improvement studies (Madaan et al. 2023; Yuan et al. 2024). Further details can be found in §3.2.

To back up the SELF-[IN]CORRECT hypothesis, we conduct experiments covering widely used LLMs (Phi-3, LLaMA series, Mixtral series, and GPT series) on a diverse set of tasks including mathematics, world knowledge acquisition, truthful question answering, and instructions following. Our investigation in §4.2 reveals a surprising finding: while there is evidence that humans find the task of discrimination simpler than generation (Alexander 2003), on various evaluated tasks we observed that LLMs are not consistently better at discriminating among previously-generated alternatives than generating initial responses.

We conduct a series of analyses to uncover the potential root causes of SELF-[IN]CORRECT. First, we investigate alternative prompt choices (§5.1) to ensure that SELF-[IN]CORRECT is not simply a result of sub-optimal prompt design. Our findings indicate that SELF-[IN]CORRECT persists with the inclusion of additional in-context learning examples and chain-of-thought demonstrations. Next, we examine the role of pre-training objectives (§5.2) and discover that SELF-[IN]CORRECT does not appear in *non*-autoregressive models (e.g., the FLAN family). Third, we discuss the apparent contradiction of our method with recent studies on self-improvement (Madaan et al. 2023; Yuan et al. 2024) (§5.3). Finally, we highlight the potential implications of our findings and a new experimental setting where more easily distinguishable incorrect options are introduced (§6).

Our contributions in this paper are two-fold:

- We develop a unified framework that facilitates the testing of both generative and discriminative capabilities of any LLM on any task;
- We conduct experiments on widely used LLMs and collected empirical evidence to support SELF-[IN]CORRECT. We also provided additional experiments to better understand SELF-[IN]CORRECT and its implications.

2 Related Work

Self-Improvement with LLMs. The concept of self-improvement existed before the LLM era. Earlier approaches employed generative adversarial networks (GANs) (Subramanian et al. 2017; Yu et al. 2017) to improve NLP systems via self-generated feedback. Welleck et al. (2023) trained a separate corrector model to iteratively refine generations.

In the era of LLM, self-improvement with self-feedback has also been studied in various forms (Pan et al. 2023; Saunders et al. 2022). Self-Instruct (Wang et al. 2023b) improves the instruction-following capabilities of pre-trained language models by bootstrapping off their own generations. Yuan et al. (2024) employs LLMs to provide rewards for their own generation. Chen et al. (2024b) uses a self-play mechanism where the LLM refines its capability by playing against instances of itself. Shinn, Labash, and Gopinath (2023) achieves self-improvement by having the model generate verbal reflection on its own outputs at inference time. Several other recent studies (Madaan et al. 2023; Liu et al. 2023a; Butt et al. 2024; Krishna 2023; Wang et al. 2023c) also adopted this idea and applied it to different tasks.

The success stories mentioned in previous paragraph show that when external ground-truth feedback is available, LLMs can effectively engage in self-improvement. Gou et al. (2024) demonstrates that LLMs can verify and correct their initial responses through interactions with various external tools. Similarly, Tyen et al. (2023) and Shinn, Labash, and Gopinath (2023) have shown that ground-truth feedback can significantly enhance LLM performance across various tasks.

However, in the absence of ground-truth feedback—where LLMs must refine their initial answers based solely on their inherent capabilities (i.e., intrinsic self-improvement)—the situation changes. Critics have argued (Huang et al. 2023; Valmeekam, Marquez, and Kambhampati 2023; Tyen et al. 2023) that the reported self-improvement on reasoning tasks may be no more effective than self-consistency (Wang et al. 2022a), and that such improvements are often a result of an inferior initial response. Our work adopts a similar intrinsic self-improvement setting, and we explore LLMs' generative and discriminative capacities beyond reasoning tasks.

Discrepancy between LLM generation and discrimination. For humans, distinguishing a good solution from a bad one is often easier than coming up with a solution from scratch (Alexander 2003). However, recent studies are starting to question if the same applies to LLMs. West et al. (2023) and Tan et al. (2024) investigated multiple NLP tasks, and showed that LLMs often struggle to understand their own outputs. To evaluate the generation and discrimination performance of LLMs, Liu et al. (2023b) conducted experiments focusing on summarization. Arora and Kambhampati (2023) and Chen et al. (2024c) conducted similar experiments in the domain of planning. Our work differs from previous research by evaluating this discrepancy on a wider range of tasks using a unified metric while also trying to uncover the reasons behind it.

Using LLMs for self-evaluation. Recent studies indicate the potential of LLM evaluation that is close to human level (Chiang and Lee 2023; Gilardi, Alizadeh, and Kubli 2023; Lin et al. 2024). However, for the task of self-evaluation, concerns have been raised by Valmeekam, Marquez, and Kambhampati (2023) and Huang et al. (2023), who pointed out that LLM encounters difficulties in self-evaluating its generation for mathematical tasks. Further research by Stechly, Valmeekam, and Kambhampati (2024), Stechly, Marquez, and Kambhampati (2023) and Valmeekam, Marquez, and Kambhampati (2023) has uncovered models' limitations on self-evaluation for tasks requiring complex reasoning and planning. Compared to these works, our work seeks to explore the efficacy of LLM self-evaluation in a broader range of tasks.

3 SELF-[IN]CORRECT

We formally define our evaluation setting (exemplified in Figure 1), and present our hypothesis.

3.1 Establishing an Evaluation Criteria to Compare Generation vs. Discrimination

Given a task T with an evaluation dataset $D = \{(x_i, y_i)\}_{i=1}^{m}$ and evaluation metric f, we use the same LLM, denoted by P_{LM}, for both generation and discrimination. For each evaluation input x_i, we first sample n candidate generations $g_1(x_i), \ldots, g_n(x_i) \sim P_{\text{LM}}(x_i)$ using the default task prompt (generation prompt). We use a low temperature during sampling to ensure the generated outputs are all highly probable.

Evaluating generation. The performance of the *generative* phase for each evaluation sample x_i is computed applying the evaluation metric f to a randomly chosen generation from the n candidate generations $G(x_i) = \{g_1(x_i), g_2(x_i), \ldots, g_n(x_i)\}$:

$$S_{\text{gen}}(x_i) = f\Big(g_{\text{rand}}(x_i), y_i\Big),$$

where g_{rand} is a randomly-sampled generation $g_{\text{rand}}(x_i) \sim G(x_i)$. The notation $f(g_j(x_i), y_i)$ represents the metric f applied to the j-th generation output for the i-th evaluation sample and $g_{\text{rand}}(x_i)$ represents one random candidate generations for that sample. Because the candidate generations for each sample are produced by sampling from the language model P_{LM} using the same hyper-parameters (temperature, top-p, etc), choosing a random candidate from $G(x_i)$ is essentially equivalent to generating an output directly from $P_{\text{LM}}(x_i)$. The overall generation performance S_{gen} is the average of $S_{\text{gen}}(x_i)$ across all samples:

$$S_{\text{gen}} = \frac{1}{m}\sum_{i=1}^{m} S_{\text{gen}}(x_i).$$

Evaluating discrimination. To assess the discrimination performance, we feed the generations back to P_{LM} and prompt it to identify the most suitable answer. For each task T, we construct a discriminative prompt $p_{\text{disc},T}$ (the prompts are available in Appendix D) and feed it the n-many generated responses. Note that the ordering of generated candidates is always random as the sampling is conducted uniformly with temperature > 0. We tried reordering the candidates before sending them for discrimination and the result remains very similar. Using few-shot prompting, we guide P_{LM} to output the label of the preferred `chosen` answer and the label `chosen` in $\{1, 2, \ldots, n\}$ is determined by greedily decoding the output of $P_{\text{LM}}(\cdot|x_{\text{disc},T}(G(x_i)))$. The discrimination performance for each sample i is quantified by:

$$S_{\text{disc}}(x_i) = f\Big(g_{\text{chosen}}(x_i), y_i\Big).$$

To derive an overall measure of discrimination performance S_{disc}, we average the individual scores $S_{\text{disc}}(x_i)$ across all samples:

$$S_{\text{disc}} = \frac{1}{m}\sum_{i=1}^{m} S_{\text{disc}}(x_i).$$

We also consider evaluating discriminative ability by calculating the absolute score of each candidate separately and selecting the best candidate. But we didn't find much difference compared to the setup here (details in Appendix D).

3.2 Hypothesis Formulation

Given the above definitions, our main hypothesis becomes easy to formalize. For any given task, denote DG-DIFF as the **difference between discrimination performance and generation performance**,

$$\text{DG-DIFF} = S_{\text{disc}} - S_{\text{gen}}.$$

Our main hypothesis is:

SELF-[IN]CORRECT. LLMs are not reliably better at discriminating among previously generated alternatives (S_{disc}) than generating initial responses (S_{gen}) and hence, DG-DIFF $= S_{\text{disc}} - S_{\text{gen}} \leq 0$.

Hypothesis testing for SELF-[IN]CORRECT. To validate SELF-[IN]CORRECT, one can apply the framework of statistical hypothesis testing (Dror et al. 2018; Sadeqi Azer et al. 2020). We treat SELF-[IN]CORRECT as the null hypothesis ($\mathbf{H}_0$) to provide an objective basis for testing In this context, the conventional wisdom that discrimination is better than generation serves as the alternative hypothesis ($\mathbf{H}_1$). To reject the null hypothesis $\mathbf{H}_0$, it must be demonstrated that DG-DIFF is a sufficiently large positive value to justify its rejection. Details of the hypothesis testing on our experimental datasets are provided in §4.1.

Design choices for hypothesis testing. An important design choice in our framework is that the candidate generations $G(x_i)$ are shared across the generative and discriminative phases. This design choice allows us to formulate the generative phase as a *random multiple choice* among pre-generated candidates. As a result, it allows a fair comparison with the discriminative phase, where the task is *using LLM for multiple choice* among the same candidates.[2]

Our framework applies the task's original metrics in both the generative and discriminative phases, which ensures consistency across assessments. By eliminating the need for human input, our framework is more scalable and cost-effective than West et al. (2023) and Zheng et al. (2023a), which depend on human annotation for discrimination. Our metrics are also closely aligned to the actual process that's employed in self-improvement literature (Shinn, Labash, and Gopinath 2023; Madaan et al. 2023), where the model is asked to choose the best answer from a list of generations. Nevertheless, we would like to mention that because generation and discrimination are two very different processes, the metrics used in this paper are only proxies to evaluate those two important capabilities.

4 Empirical Support for SELF-[IN]CORRECT

In this section, we describe our experimental setup (§4.1) and lay out the main findings (§4.2).

[2]Under this setup, the generative and discriminative phases will always have the same upper/lower bound of possible scores.

Task	Split	#Eval	#Shots	Task Type	Metric $f(.)$	Metric scale
GSM8K	Test	1319	2	Math Word Problem	Accuracy	$[0, 100]$
TriviaQA	Val	17944	2	Question Answering	Accuracy	$[0, 100]$
MT-Bench	Test	160	3	Instruction Following	GPT-4 score	$[0, 10]$
TruthfulQA	Val	817	2	Question Answering	GPT-judge	$[0, 100]$

Table 1: Configuration of experimental tasks. "Split" specifies which subset the data originates from. "#Eval" indicates the number of instances used for evaluation. "#Shots" specifies the number of few-shot examples employed for evaluation. To evaluate TruthfulQA generations, we follow Lin, Hilton, and Evans (2022) and develop two "GPT-judges" by fine-tuning GPT-3 models with provided data.

4.1 Experimental Setup

Tasks. A summary of the tasks we evaluate on is provided in Table 1. We assess our hypothesis on a diverse set of tasks including GSM8K (Cobbe et al. 2021) for math, TriviaQA (Joshi et al. 2017) for world knowledge, TruthfulQA (Lin, Hilton, and Evans 2022) for truthfulness in question answering, and MT-Bench (Zheng et al. 2023a) for instruction following. These represent a diverse set of benchmarks used to evaluate LLMs across various domains. For TriviaQA, we use the `rc.nocontext` setup, which means the model relies solely on its parametric knowledge to answer the question correctly without accompanying context or documents. For TruthfulQA, we use the generation setup, where the model generates responses to a set of questions. The metrics scale for MT-Bench is 0-10 (Zheng et al. 2023a).

Task metrics. The list of task-specific metrics $f(.)$ is provided in Table 1. The evaluation for GSM8K, TriviaQA and TruthfulQA is conducted using `lm-evaluation-harness`[3], which provides a standardized framework for assessing model performance across benchmarks. The evaluation for MT-Bench is done with `llm_judge`[4], which use GPT-4 score (Zheng et al. 2023a) to score the generated answer from models. We do not test GPT-4-turbo on MT-Bench to avoid self-evaluation bias (He et al. 2023). To evaluate TruthfulQA, we follow Lin, Hilton, and Evans (2022) and develop two "GPT-judges" by fine-tuning GPT-3 models[5] with provided data . Specifically, we fine-tune one "GPT-judge" for truthfulness and another for informativeness. Finally, we report the percentage of answers that are both truthful and informative as the final metric for TruthfulQA.

Hypothesis Testing for SELF-[IN]CORRECT Across Tasks. We apply a one-sided McNemar's Test (Mcnemar 1947) for GSM8K, TriviaQA, and TruthfulQA to calculate p-values and assess statistical significance, as this test is ideal for binary outcome comparisons. For MT-Bench, we use the Wilcoxon signed-rank test (Wilcoxon 1945) because it handles categorical data and does not assume a normal distribution. Further details on our test selection and hypothesis testing methodology are provided in Appendix G.

Handling failure modes during evaluation. During the evaluation of the discrimination phase, if the model's output does not conform to the expected format (i.e., integers indicating the selected answer), we consider it a failure. While such an output would receive a score of 0 in the generative setting, we take a more lenient approach for the discrimination phase. In these cases, we assign the model the score of the lowest-performing generated answer (according to our metric f) among the other candidate answers: $S_{\text{disc}}(x_i) = \min_{g(x_i) \in G(x_i)} f(g(x_i), y_i)$. We also try to make the discriminator output one of the answers directly in the case of a failure, hoping that bypassing this extra step of identifying the multiple-choice options would simplify the discrimination phase. However, we observe an increased percentage of invalid discrimination output with similar discrimination performance (see Appendix C for more details). In our experiments, we observe that the average rate of invalid responses remained low (often less than 5%). Given that there is also a small proportion of invalid outputs in the generation phase that wouldn't get any credit when selected, we believe that the occurrence of invalid discrimination outputs does not significantly impact our overall findings.

Models. We employ a range of models including Phi-3-mini-4k-Instruct (Abdin et al. 2024), LLaMA-2 Base models (7B, 13B, and 70B), LLaMA-2 Chat models (7B, 13B and 70B), LLaMa-3 Base models (8B and 70B), LLaMa-3 Instruct models (8B and 70B), Mixtral $8 \times$ 7B-Instruct-v0.1, GPT-3.5-turbo and GPT-4-turbo.[6] For the evaluation of each model, we adapt our prompts to be compatible with the keywords used in their [pre-]training. For example, when prompting LLaMA-2 Chat models we use `<SYS>`, `<INST>` keywords to indicate system and instruction prompts.

Model hyper-parameters. During the generation phase, we use the default hyperparameter specified in `lm-eval-harness` for all tasks, except for temperature, which we have adjusted to 0.7. We use an above 0 temperature to obtain distinct generations upon multiple rounds of sampling. At the same time, during the discrimination phase, we set the temperature to 0 to avoid any randomness.

[3]https://github.com/EleutherAI/lm-evaluation-harness

[4]https://github.com/lm-sys/FastChat/tree/main/fastchat/llm_judge

[5]The original "GPT-judges" were fine-tuned with `curie` models which are no longer available for fine-tuning. Therefore, we use `davinci-002` which is larger than `curie`.

[6]We use GPT-3.5-turbo-0125 and GPT-4-0125-preview.

4.2 Main Findings

On a dominant majority of experiments, SELF-[IN]CORRECT is not rejected. Based on the results in Table 2, in 54 out of 56 experiments, the p-value exceeded the significance level (0.05), leading to the failure to reject the SELF-[IN]CORRECT hypothesis. In fact, DG-DIFF is generally small or negative across both pre-trained models and aligned models. To test the effect of prompt variations, we conduct an ablation experiment in Appendix E and find that these variations do not significantly affect DG-DIFF. Although in few cases (2 out of 56) the p-value is high enough to reject our hypothesis (p-value > 0.05), such as LLaMA-2-70B and GPT-3.5-turbo on TriviaQA, DG-DIFF remains quite small in such cases. We would also like to point out that these cases start with high generative accuracy and the *relative* differential in discrimination is quite minimal. All these observations lend support for SELF-[IN]CORRECT.

Instruction fine-tuned models went through both instruction-tuning and RLHF alignment while the base models are only pre-trained with the autoregressive objective. It is reasonable to expect that instruction-tuned models would exhibit better performance in the discrimination phase as instruction-tuning is shown to make models better at solving a variety of tasks. Furthermore, classification tasks (that resemble our discrimination setup) are well-represented in most instruction-tuning datasets (Wang et al. 2022b; Bach et al. 2022; Longpre et al. 2023). However, our empirical findings do not support it.

Stronger models tend to be better at discrimination (larger DG-DIFF). Our research has observed an interesting trend: an increase in DG-DIFF seems to correlate with model capacity. This pattern is particularly pronounced among models in the same category (base models, fine-tuned models, and proprietary models developed by OpenAI). We also want to emphasize that some of the strongest models we tested—specifically LLaMa-3-70B, LLaMa-3-70B-Instruct, GPT-3.5-turbo, and GPT-4-turbo—show a positive DG-DIFF across nearly all evaluated tasks, though the gap remains small enough for SELF-[IN]CORRECT to still hold. We hypothesize that this is because weaker models have limited discrimination capabilities. Similar observations on the weaker models' discrimination capability have been reported in other studies (Saunders et al. 2022; Kadavath et al. 2022).

5 Further Analysis of SELF-[IN]CORRECT

In this section, we outline experiments designed to provide further analysis of SELF-[IN]CORRECT.

5.1 Better Discrimination via Prompt-Engineering

One might argue our current prompting setup doesn't fully capitalize on the model's capacity for discrimination. To make sure SELF-[IN]CORRECT isn't an artifact of poor prompt engineering, we conduct additional experiments with LLaMA-2 Chat models on GSM8K, TriviaQA, and MT-Bench as their DG-DIFF on those tasks is mostly negative.

More in-context learning examples helps discrimination, though DG-DIFF remains small or negative. Increasing the number of in-context learning (ICL) demonstrations is shown to improve performance (Brown et al. 2020). Is it possible that increasing the number of ICL examples in the discrimination phase will improve it, so much that DG-DIFF becomes consistently positive? To evaluate the effect of increasing ICL examples, we conduct experiments where the number of ICL examples (#Shots) during the discrimination stage is doubled or tripled relative to the baseline in Table 1. Note that we didn't triple the number of ICL examples for MT-Bench because it exceeds the context length for LLaMa-2 Chat models (4096 tokens). The results, presented in Table 3, indicate that while increasing the number of ICL examples tends to increase DG-DIFF, it remains small or negative. Furthermore, the performance improvement from adding ICL examples does not exhibit a consistent monotonic trend.

Chain-of-thought rational shows minimal impact on DG-DIFF. Recently, Stechly, Valmeekam, and Kambhampati (2024) pointed out that LLM evaluation also involves multi-step reasoning. To help with the reasoning in the discrimination phase, we add chain-of-thought rationals in the few-shot examples while keeping the number of examples constant. For GSM8K, we do not report anything since our default evaluation already contains rationales for answer selection. For TriviaQA, the CoT rationales explain the logic behind choosing an option. For MT-Bench, we supplement explanations for preferring one answer over another. A comparison between our prompts (w/ and w/o CoT rationales) is available in Appendix A. As shown in Table 3, the inclusion of CoT rational only shows minimal impact.

5.2 The Role of Objectives: Does Autoregressive Pre-training Explain our Results?

The majority of modern LLMs are pre-trained with an autoregressive objective. Recent studies suggest that autoregressive objectives used during pre-training may have unexpected impacts on LLM behavior (McCoy et al. 2023). Since the pre-training process of autoregressive models is more similar to generation than discrimination, we hypothesize SELF-[IN]CORRECT *is also partially caused by the use of autoregressive pre-training objective*.

To test this hypothesis, we evaluate Flan-T5-XXL (11B) and Flan-UL2 (20B) on the same tasks listed in Table 2, as these are the only prominent open-source non-autoregressive models available to the best of our knowledge. Flan-T5-XXL is pre-trained using a span corruption objective, where the loss is only calculated on the corrupted span (Raffel et al. 2020). Flan-UL2 (Chung et al. 2022) is pre-trained using mixture-of-denoisers that combines multiple denoising objective functions. Our findings, detailed in Table 4, reveal their DG-DIFF across all tasks are positive except for Flan-T5-XXL on MT-Bench. In fact, both models exhibit significantly higher DG-DIFF and even more significantly higher *relative* DG-DIFF compared to the autoregressive models we tested in Table 2. Moreover, for both TriviaQA and TruthfulQA, the SELF-[IN]CORRECT hypothesis is rejected. This outcome lends empirical support to the hypothesis that SELF-[IN]CORRECT could be related to autoregressive pre-training.

		GSM8K		TriviaQA		MT-Bench		TruthfulQA	
		DG-DIFF	p-value	DG-DIFF	p-value	DG-DIFF	p-value	DG-DIFF	p-value
Base Models	**LLaMA-2 7B**	$-0.6_{(9.2\rightarrow8.6)}$	-	$-16.9_{(37.1\rightarrow20.2)}$	-	$-0.09_{(3.34\rightarrow3.25)}$	-	$-4.7_{(30.5\rightarrow25.8)}$	-
	LLaMA-2 13B	$0.0_{(16.8\rightarrow16.8)}$	0.50	$1.4_{(45.2\rightarrow46.6)}$	0.07	$-0.12_{(4.15\rightarrow4.03)}$	-	$2.1_{(26.8\rightarrow28.9)}$	0.10
	LLaMA-2 70B	$2.2_{(44.0\rightarrow46.2)}$	0.12	$3.2_{(53.2\rightarrow56.4)}$	$0.00^{\triangle}$	$-0.12_{(4.87\rightarrow4.75)}$	-	$0.5_{(28.9\rightarrow29.4)}$	0.40
	LLaMA-3 8B	$-3.6_{(38.6\rightarrow35.0)}$	-	$-2.3_{(45.4\rightarrow43.1)}$	-	$0.06_{(3.47\rightarrow3.53)}$	0.42	$0.2_{(27.2\rightarrow27.4)}$	0.47
	LLaMA-3 70B	$1.1_{(77.7\rightarrow78.8)}$	0.25	$1.1_{(64.2\rightarrow65.3)}$	0.09	$0.14_{(5.32\rightarrow5.46)}$	0.36	$0.8_{(36.8\rightarrow37.6)}$	0.37
Aligned Models	**Phi-3-mini-3.8B-Instruct**	$-0.2_{(77.9\rightarrow77.7)}$	-	$0.7_{(22.1\rightarrow22.9)}$	0.11	$-0.08_{(7.33\rightarrow7.25)}$	-	$-0.2_{(26.3\rightarrow26.1)}$	-
	LLaMA-2 7B-Chat	$-2.8_{(20.4\rightarrow17.6)}$	-	$-0.1_{(16.1\rightarrow16.0)}$	-	$-0.13_{(5.45\rightarrow5.32)}$	-	$1.4_{(48.8\rightarrow50.2)}$	0.20
	LLaMA-2 13B-Chat	$-5.5_{(28.3\rightarrow22.8)}$	-	$0.0_{(25.5\rightarrow25.5)}$	-	$-0.51_{(5.67\rightarrow5.16)}$	-	$-0.1_{(44.9\rightarrow44.8)}$	-
	LLaMA-2 70B-Chat	$-5.9_{(42.5\rightarrow36.6)}$	-	$-1.6_{(47.8\rightarrow46.2)}$	-	$-0.17_{(6.65\rightarrow6.48)}$	-	$0.9_{(48.6\rightarrow49.5)}$	0.31
	LLaMA-3 8B-Instruct	$1.0_{(76.9\rightarrow77.9)}$	0.29	$0.6_{(48.7\rightarrow49.3)}$	0.26	$0.14_{(6.31\rightarrow6.45)}$	0.32	$-0.6_{(50.1\rightarrow49.5)}$	-
	LLaMA-3 70B-Instruct	$0.6_{(92.2\rightarrow92.8)}$	0.31	$1.1_{(64.2\rightarrow65.3)}$	0.11	$0.19_{(7.60\rightarrow7.79)}$	0.23	$-1.1_{(56.2\rightarrow55.1)}$	-
	Mixtral-8x7B-Instruct	$1.3_{(59.6\rightarrow60.9)}$	0.37	$-3.4_{(58.8\rightarrow55.4)}$	-	$-0.20_{(7.39\rightarrow7.19)}$	-	$-0.4_{(61.1\rightarrow60.7)}$	-
	GPT-3.5-turbo	$1.1_{(75.3\rightarrow76.4)}$	0.37	$2.1_{(67.1\rightarrow69.2)}$	$0.01^{\triangle}$	$0.17_{(7.74\rightarrow7.91)}$	0.26	$0.4_{(65.7\rightarrow66.1)}$	0.41
	GPT-4-turbo	$0.7_{(93.6\rightarrow94.3)}$	0.39	$0.2_{(79.9\rightarrow80.1)}$	0.40	–	-	$1.7_{(77.4\rightarrow79.1)}$	0.09
	Task Avg.	-0.76	-	-0.99	-	-0.06	-	0.06	-

Table 2: Performance change defined as DG-DIFF $:= S_{\text{disc}} - S_{\text{gen}}$, with p-values indicating the likelihood that the observed difference is due to chance, for various mainstream LLMs on different tasks. The generation performance and discriminative performance are shown as subscript: $(S_{\text{gen}} \rightarrow S_{\text{disc}})$. p-values are calculated only when DG-DIFF is greater than or equal to 0. A red p-value marked with $\triangle$ signifies a value less than 0.05. **For the majority of our results, DG-DIFF is small or negative, indicating similar or worse performance in the discrimination phase, and the p-value for 54/56 experiments is less than 0.05, meaning SELF-[IN]CORRECT is not rejected.**

Model	LLaMA-2 7B Chat			LLaMA-2 13B Chat			LLaMA-2 70B Chat		
Setup	+2×#ICL	+3×#ICL	+CoT	+2×#ICL	+3×#ICL	+CoT	+2×#ICL	+3×#ICL	+CoT
GSM8K	-1.4	0.1	-	-5.9	-6.8	-	-5.8	-3.9	-
TriviaQA	-0.4	0.2	-0.3	0.1	0.1	-0.3	-1.7	-0.5	-1.8
MT-Bench	-0.09	-	-0.06	-0.53	-	-0.41	-0.19	-	-0.18

Table 3: DG-DIFF upon various modifications with LLaMA-2 Chat models. "+ 2 × #ICL" means doubling the number of in-context demonstrations during the discrimination phase. "+ 3 × #ICL" means tripling the number of in-context demonstrations. "+ CoT" stands for adding Chain-of-Thought rationales for the few-shot examples. **Extra prompt-engineering techniques during discrimination do not consistently close the performance gap.**

	DG-DIFF$_{(S_{\text{gen}} \rightarrow S_{\text{disc}})}$	
	Flan-T5 XXL	Flan-UL2
GSM8K	$1.3_{(13.3\rightarrow14.4)}$	$0.5_{(21.6\rightarrow22.1)}$
TriviaQA	$5.8_{(28.7\rightarrow34.5)}$	$4.2_{(52.7\rightarrow56.9)}$
MT-Bench	$-0.06_{(2.02\rightarrow1.96)}$	$0.16_{(1.98\rightarrow2.14)}$
TruthfulQA	$6.0_{(20.1\rightarrow26.1)}$	$4.8_{(31.3\rightarrow36.1)}$

Table 4: Flan-T5-XXL and Flan-UL2 tested on the same setup as Table 2. DG-DIFF for all models across all tasks are positive except for Flan-T5-XXL on MT-Bench. **Both models demonstrate significantly higher DG-DIFF compared to autoregressive models.**

It is also important to note that the pre-training processes for these two model classes differ from autoregressively pre-trained counterparts beyond the objective function. For example, Flan-T5 is pre-trained with most inputs provided, except for the corrupted spans. Furthermore, the datasets used for pre-training these LLMs can vary. Therefore, when uncovering the underlying reason why SELF-[IN]CORRECT does not occur on Flan-T5 and Flan-UL2, caution should be exercised before drawing definitive conclusions.

5.3 Do Prior Findings in Self-Refinement Contradict SELF-[IN]CORRECT?

The process of self-refine involves utilizing the same LLM to provide feedback for its own generation and using the feedback to refine the generation. Both Huang et al. (2023) and Madaan et al. (2023) suggested LLMs can self-refine on tasks other than reasoning. Does this contradict our assertions?

We replicated the experiment outlined in Madaan et al. (2023) and observed the following:

(1) **For some evaluated tasks, certain aspects can be exploited for artificially amplifying task performance without actually improving with the feedback.** For example, on the task of constrained generation, where the objective is to generate sentences containing specific keywords, self-refine with LLMs often leads to progressively longer sentences that

simply extend previous ones. Thus, even if the refined sentences do not incorporate new keywords and continue to grow longer (often, ignoring the feedback from the prior round), the task performance still shows a monotonic improvement. A more detailed explanation of this behavior across additional tasks is provided in Table 5. To further illustrate our point, an example question and model output from the acronym generation task can be found in Figure 6 in Appendix B, and another example from the constraint generation task is presented in Figure 7 in the same appendix.

(2) **For some evaluated tasks, the evaluation score assigned by the model for each iteration of self-refine is not monotonically increasing.** We use the same model involved in self-refinement to evaluate each refined output. Ideally, the scores would improve with refinement, but for tasks like acronym generation and dialogue response, the model often assigns lower scores to refined outputs. This suggests that the observed improvement may be due to lower initial output quality, as noted in Huang et al. (2023).

(3) **Quantifying the percentage of times models prefer self-refined subsequent generations to the previous generation, a marginal preference for self-refined generation was observed.** We used the same models to discriminate between previous generations and self-refined subsequent generations for tasks referenced in Madaan et al. (2023), thus extending the evaluation of SELF-[IN]CORRECT to a broader range of real-world tasks beyond reasoning. Our results in Table 5 indicate that models prefer self-refined generations only around 54% of the time, meaning on those tasks LLMs are still not consistently better at discriminating among previously-generated alternatives than generating initial responses.

6 Further Discussion

SELF-[IN]CORRECT likely poses a barrier for continued progress in self-rewarding LLMs. Few recent works like Self-Rewarding Language Models (Yuan et al. 2024) generate preference pairs consisting of an instruction prompt x, a winning response y_win, and a losing response y_lose to facilitate self-reward for instruction-following fine-tuning. While their setup also involves discriminating between previously generated outputs, our findings do not challenge its effectiveness. Their method selects winning and losing pairs from a larger set of generations, with the key factor being the correct ordering of these pairs. This setup simplifies the task, especially when straightforward heuristics like choosing the highest and lowest-scoring responses exist. In contrast, our approach requires the model to identify the single best generation, demanding a much finer level of granularity.

However, an interesting pattern from Yuan et al. (2024) is that there seem to be diminishing returns after a few iterations of self-rewarding. We hypothesize this may be linked to SELF-[IN]CORRECT because if the overall ability of LLMs to discriminate is inferior to their ability to generate, it becomes challenging to engage in a virtuous cycle that simultaneously enhances the model's capability to follow instructions and generate self-rewards.

Controlled Modification of Experimental Setting with Simplified Distractors Here we consider the extent to which SELF-[IN]CORRECT may hold. For example, is SELF-[IN]CORRECT potentially a fundamental limitation of LLMs pre-trained with an autoregressive objective, or can a change in data distribution alter the outcome? To address this question, we conduct experiments in an unconventional setting that simplifies the discrimination phase by substituting incorrect candidates with simpler ones for discrimination.

The experiments are conducted on TriviaQA and GSM8K. TriviaQA contains a wide range of answer categories, including names, locations, historical events, etc. For this dataset, we simplify the discrimination phase by substituting incorrect answer generations for question A with correct answer generations from another question B (left panel in Figure 8, Appendix F). As for GSM8K, we create simplified distractors by randomly multiplying or dividing incorrect generated answers by 100 (right panel in Figure 8, Appendix F).

Figure 9 in Appendix F clearly shows that simplifying the incorrect candidates improves DG-DIFF. For TriviaQA, S_{disc} exceeds S_{gen} by a large margin. For GSM8K, all models tested also demonstrate improved DG-DIFF.

7 Limitations

Challenges in controlled study of LLMs in relation to SELF-[IN]CORRECT. One limitation of our research stems from the difficulty in measuring the impact of pre-training data and pre-training objectives. The vast amount of pre-training data makes it hard to evaluate its effect, leaving important aspects underexplored.

Potential influence of lengthier discrimination prompt on SELF-[IN]CORRECT. The prompt used in the discrimination phase is inherently lengthier than the generation prompt as it also includes the generated candidate answers. This increase in length may pose challenges to the model's processing capabilities. Investigating the impact of prompt length is complex as simply adding superfluous content to lengthen the generation or discrimination prompt might unintentionally influence the outcomes. Therefore, we highlight this area for further exploration to better understand the implications of prompt length for SELF-[IN]CORRECT.

Limitations in experimental scope Another limitation of our study is the scope of our experiments. While we tested SELF-[IN]CORRECT across multiple tasks and domains using prominent LLMs, expanding to more models and tasks could further validate SELF-[IN]CORRECT.

We want to note while the current results support SELF-[IN]CORRECT, we are *not* claiming that LLMs can *never* be better at discrimination than generation. Some studies (Welleck et al. 2023; Chen et al. 2024a) suggest that fine-tuning specifically on refinement data can improve discrimination capabilities, though this may come at the expense of the model's generality. Whether it is possible to train a model that maintains generality while excelling at discrimination remains an open research question.

In addition, our discrimination setup is designed to be simple, allowing our method to be directly applied to a wide

Task	Issues	Detailed Explanation	Pref.%
Sentiment Reversal	Lack of Reasoning	Refinements simply make the sentiment more and more positive	58.7%
Dialogue Response Generation	Reward Inconsistency	Reward assigned by LLMs doesn't increase monotonically	52.4%
Code Readability Improvement	Lack of Reasoning	Refinements simply makes variable names longer and more descriptive	53.3%
Acronym Generation	Reward Inconsistency	Reward assigned by LLMs doesn't increase monotonically	46.5%
Constrained Generation	Lack of Reasoning	Refinements simply extends previous generation	54.7%

Table 5: **Explanation for some of the issues on tasks that Madaan et al. (2023) tested and the percentage of times the model prefers self-refined subsequent generations than previous generations.** GSM8K isn't included here because it didn't get much improvement through self-refine in the original paper. Code optimization isn't included either due to the complexity of running experiments.

range of tasks while helping us better understand the inherent characteristics and challenges of LLMs. Exploring more complex techniques like problem decomposition and answer verification to enhance discrimination is beyond the scope of this paper.

Challenges in determining model's preference Determining a model's preference over several candidate generations can be challenging due to various biases (Alzahrani et al. 2024; Wang et al. 2024). Following the methodology used in other self-improvement studies (Yuan et al. 2024), we employ *LLM-as-a-Judge prompting* (Zheng et al. 2023b) to elicit answer choice from the model. It is conceivable that the LLMs we examine can be biased toward certain answer options or different answer formats (e.g., labels of A/B/C/D or [1]/[2]/[3]/[4]). Another method that we did not explore is ranking candidate answers based on the LLM's assigned probability for each answer text. However, it is worth noting that this approach can also be biased by factors like the text fluency from the pre-training data.

Limited focus on other stages of self-improvement Self-improvement involves multiple stages, such as self-discrimination, critique generation, and generating additional answers after self-evaluation. Our focus is mainly on the first stage, with less emphasis on others. However, it is generally agreed (Huang et al. 2023; Tyen et al. 2023) that for LLMs to succeed at other stages reliably, they must first excel at the initial self-discrimination stage.

Even within the first stage, approaches can vary; some generate a single answer and decide if another is needed, whereas our work explores generating multiple answers followed by discrimination. However, we believe that overall success in this stage is ultimately dependent on the model's discrimination capability.

8 Conclusion

We focused on the question of whether language models are strictly better at discriminating their prior generations vs. generating responses directly. We proposed a metric for comparing these capabilities and used it to evaluate several current LLMs. For those models and tasks, we do not observe that discrimination is reliably better than generation, in fact, we often observed it was worse. These results raise concerns about the potential for LLM self-improvement on ***any*** task.

Acknowledgments

This work is in-part supported by ONR grant N00014-241-2089 and a generous gifts from Amazon. We are also grateful to the broader JHU CLSP community and our anonymous reviewers for their support and constructive feedback. The GPUs for conducting experiments were provided by the DSAI cluster.

References

Abdin, M.; Jacobs, S. A.; Awan, A. A.; Aneja, J.; Awadallah, A.; Awadalla, H.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; Benhaim, A.; Bilenko, M.; Bjorck, J.; Bubeck, S.; Cai, Q.; Cai, M.; Mendes, C. C. T.; Chen, W.; Chaudhary, V.; Chen, D.; Chen, D.; Chen, Y.-C.; Chen, Y.-L.; Chopra, P.; Dai, X.; Giorno, A. D.; de Rosa, G.; Dixon, M.; Eldan, R.; Fragoso, V.; Iter, D.; Gao, M.; Gao, M.; Gao, J.; Garg, A.; Goswami, A.; Gunasekar, S.; Haider, E.; Hao, J.; Hewett, R. J.; Huynh, J.; Javaheripi, M.; Jin, X.; Kauffmann, P.; Karampatziakis, N.; Kim, D.; Khademi, M.; Kurilenko, L.; Lee, J. R.; Lee, Y. T.; Li, Y.; Li, Y.; Liang, C.; Liden, L.; Liu, C.; Liu, M.; Liu, W.; Lin, E.; Lin, Z.; Luo, C.; Madan, P.; Mazzola, M.; Mitra, A.; Modi, H.; Nguyen, A.; Norick, B.; Patra, B.; Perez-Becker, D.; Portet, T.; Pryzant, R.; Qin, H.; Radmilac, M.; Rosset, C.; Roy, S.; Ruwase, O.; Saarikivi, O.; Saied, A.; Salim, A.; Santacroce, M.; Shah, S.; Shang, N.; Sharma, H.; Shukla, S.; Song, X.; Tanaka, M.; Tupini, A.; Wang, X.; Wang, L.; Wang, C.; Wang, Y.; Ward, R.; Wang, G.; Witte, P.; Wu, H.; Wyatt, M.; Xiao, B.; Xu, C.; Xu, J.; Xu, W.; Yadav, S.; Yang, F.; Yang, J.; Yang, Z.; Yang, Y.; Yu, D.; Yuan, L.; Zhang, C.; Zhang, C.; Zhang, J.; Zhang, L. L.; Zhang, Y.; Zhang, Y.; Zhang, Y.; and Zhou, X. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. arXiv:2404.14219.

Alexander, P. A. 2003. The development of expertise: The journey from acclimation to proficiency. In *Educational researcher*.

Alzahrani, N.; Alyahya, H. A.; Alnumay, Y.; Alrashed, S.; Alsubaie, S.; Almushaykeh, Y.; Mirza, F.; Alotaibi, N.; Altwairesh, N.; Alowisheq, A.; Bari, M. S.; and Khan, H. 2024. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards. arXiv:2402.01781.

Arora, D.; and Kambhampati, S. 2023. Learning and Leveraging Verifiers to Improve Planning Capabilities of Pre-trained Language Models. *CoRR*.

Bach, S. H.; Sanh, V.; Yong, Z. X.; Webson, A.; Raffel, C.; Nayak, N. V.; Sharma, A.; Kim, T.; Bari, M. S.; Févry, T.; Alyafeai, Z.; Dey, M.; Santilli, A.; Sun, Z.; Ben-David, S.; Xu, C.; Chhablani, G.; Wang, H.; Fries, J. A.; Al-shaibani, M. S.; Sharma, S.; Thakker, U.; Almubarak, K.; Tang, X.; Jiang, M. T.-J.; and Rush, A. M. 2022. PromptSource: An Integrated Development Environment and Repository for Natural Language Prompts. *ArXiv*, abs/2202.01279.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* (NeurIPS).

Butt, N.; Manczak, B.; Wiggers, A.; Rainone, C.; Zhang, D.; Defferrard, M.; and Cohen, T. 2024. CodeIt: Self-Improving Language Models with Prioritized Hindsight Replay. *arXiv preprint arXiv:2402.04858*.

Chen, K.; Wang, C.; Yang, K.; Han, J.; Hong, L.; Mi, F.; Xu, H.; Liu, Z.; Huang, W.; Li, Z.; Yeung, D.-Y.; Shang, L.; Jiang, X.; and Liu, Q. 2024a. Gaining Wisdom from Setbacks: Aligning Large Language Models via Mistake Analysis. arXiv:2310.10477.

Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; and Gu, Q. 2024b. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. *CoRR*.

Chen, Z.; White, M.; Mooney, R.; Payani, A.; Su, Y.; and Sun, H. 2024c. When is Tree Search Useful for LLM Planning? It Depends on the Discriminator. arXiv:2402.10890.

Chiang, D. C.; and Lee, H. 2023. Can Large Language Models Be an Alternative to Human Evaluations? In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *ACL*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, E.; Wang, X.; Dehghani, M.; Brahma, S.; Webson, A.; Gu, S. S.; Dai, Z.; Suzgun, M.; Chen, X.; Chowdhery, A.; Narang, S.; Mishra, G.; Yu, A.; Zhao, V. Y.; Huang, Y.; Dai, A. M.; Yu, H.; Petrov, S.; Chi, E. H.; Dean, J.; Devlin, J.; Roberts, A.; Zhou, D.; Le, Q. V.; and Wei, J. 2022. Scaling Instruction-Finetuned Language Models. *CoRR*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems.

Corder, S. P. 1967. The significance of learner's errors.

Dror, R.; Baumer, G.; Shlomov, S.; and Reichart, R. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 1383–1392.

Gilardi, F.; Alizadeh, M.; and Kubli, M. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*.

Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; and Chen, W. 2024. CRITIC: Large Language Models Can Self-Correct with Tool-Interactive Critiquing. arXiv:2305.11738.

He, T.; Zhang, J.; Wang, T.; Kumar, S.; Cho, K.; Glass, J.; and Tsvetkov, Y. 2023. On the Blind Spots of Model-Based Evaluation Metrics for Text Generation. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *ACL*.

Huang, J.; Chen, X.; Mishra, S.; Zheng, H. S.; Yu, A. W.; Song, X.; and Zhou, D. 2023. Large Language Models Cannot Self-Correct Reasoning Yet.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *ACL*.

Kadavath, S.; Conerly, T.; Askell, A.; Henighan, T.; Drain, D.; Perez, E.; Schiefer, N.; Hatfield-Dodds, Z.; DasSarma, N.; Tran-Johnson, E.; Johnston, S.; El-Showk, S.; Jones, A.; Elhage, N.; Hume, T.; Chen, A.; Bai, Y.; Bowman, S.; Fort, S.; Ganguli, D.; Hernandez, D.; Jacobson, J.; Kernion, J.; Kravec, S.; Lovitt, L.; Ndousse, K.; Olsson, C.; Ringer, S.; Amodei, D.; Brown, T.; Clark, J.; Joseph, N.; Mann, B.; McCandlish, S.; Olah, C.; and Kaplan, J. 2022. Language Models (Mostly) Know What They Know. arXiv:2207.05221.

Krishna, S. 2023. On the Intersection of Self-Correction and Trust in Language Models. arXiv:2311.02801.

Lin, S.; Hilton, J.; and Evans, O. 2022. TruthfulQA: Measuring How Models Mimic Human Falsehoods. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *ACL*. Association for Computational Linguistics.

Lin, Z.; Gou, Z.; Liang, T.; Luo, R.; Liu, H.; and Yang, Y. 2024. CriticBench: Benchmarking LLMs for Critique-Correct Reasoning. arXiv:2402.14809.

Liu, J.; Pasunuru, R.; Hajishirzi, H.; Choi, Y.; and Celikyilmaz, A. 2023a. Crystal: Introspective Reasoners Reinforced with Self-Feedback. *arXiv preprint arXiv:2301.04921*.

Liu, Y.; Fabbri, A. R.; Chen, J.; Zhao, Y.; Han, S.; Joty, S.; Liu, P.; Radev, D.; Wu, C.; and Cohan, A. 2023b. Benchmarking Generation and Evaluation Capabilities of Large Language Models for Instruction Controllable Summarization. *CoRR*.

Longpre, S.; Hou, L.; Vu, T.; Webson, A.; Chung, H. W.; Tay, Y.; Zhou, D.; Le, Q. V.; Zoph, B.; Wei, J.; et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Welleck, S.; Majumder, B. P.; Gupta, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *CoRR*.

Mayo, D. G. 1996. *Error and the growth of experimental knowledge*. University of Chicago Press.

McCoy, R. T.; Yao, S.; Friedman, D.; Hardy, M.; and Griffiths, T. L. 2023. Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve. *CoRR*.

Mcnemar, Q. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12: 153–157.

Pan, L.; Saxon, M. S.; Xu, W.; Nathani, D.; Wang, X.; and Wang, W. Y. 2023. Automatically Correcting Large Language Models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research* (JMLR).

Sadeqi Azer, E.; Khashabi, D.; Sabhwawal, A.; and Roth, D. 2020. Not All Claims are Created Equal: Choosing the Right Approach to Assess Your Hypotheses. In *Annual Meeting of the Association for Computational Linguistics* (ACL).

Saunders, W.; Yeh, C.; Wu, J.; Bills, S.; Ouyang, L.; Ward, J.; and Leike, J. 2022. Self-critiquing models for assisting human evaluators. arXiv:2206.05802.

Shinn, N.; Labash, B.; and Gopinath, A. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. In *NeuralPS*.

Stechly, K.; Marquez, M.; and Kambhampati, S. 2023. GPT-4 Doesn't Know It's Wrong: An Analysis of Iterative Prompting for Reasoning Problems. *CoRR*.

Stechly, K.; Valmeekam, K.; and Kambhampati, S. 2024. On the Self-Verification Limitations of Large Language Models on Reasoning and Planning Tasks. *CoRR*.

Subramanian, S.; Rajeswar, S.; Dutil, F.; Pal, C.; and Courville, A. C. 2017. Adversarial Generation of Natural Language. In *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017*, 241–251. Association for Computational Linguistics.

Tan, Z.; Wei, L.; Wang, J.; Xie, X.; and Huang, W. 2024. Can I understand what I create? Self-Knowledge Evaluation of Large Language Models. arXiv:2406.06140.

Tyen, G.; Mansoor, H.; Chen, P.; Mak, T.; and Carbune, V. 2023. LLMs cannot find reasoning errors, but can correct them! *CoRR*.

Valmeekam, K.; Marquez, M.; and Kambhampati, S. 2023. Can Large Language Models Really Improve by Self-critiquing Their Own Plans? *CoRR*.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023a. Voyager: An Open-Ended Embodied Agent with Large Language Models. arXiv:2305.16291.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; and Zhou, D. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023b. Self-Instruct: Aligning Language Model with Self Generated Instructions. In *Annual Meeting of the Association for Computational Linguistics* (ACL).

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2023c. Self-Instruct: Aligning Language Model with Self Generated Instructions. *CoRR*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; Li, T.; Ku, M.; Wang, K.; Zhuang, A.; Fan, R.; Yue, X.; and Chen, W. 2024. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. arXiv:2406.01574.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; Kordi, Y.; Mirzaei, A.; Arunkumar, A.; Ashok, A.; Dhanasekaran, A. S.; Naik, A.; Stap, D.; Pathak, E.; Karamanolakis, G.; Lai, H. G.; Purohit, I.; Mondal, I.; Anderson, J.; Kuznia, K.; Doshi, K.; Patel, M.; Pal, K. K.; Moradshahi, M.; Parmar, M.; Purohit, M.; Varshney, N.; Kaza, P. R.; Verma, P.; Puri, R. S.; Karia, R.; Sampat, S. K.; Doshi, S.; Mishra, S.; Reddy, S.; Patro, S.; Dixit, T.; Shen, X.; Baral, C.; Choi, Y.; Smith, N. A.; Hajishirzi, H.; and Khashabi, D. 2022b. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ Tasks. In *Conference on Empirical Methods in Natural Language Processing* (EMNLP).

Welleck, S.; Lu, X.; West, P.; Brahman, F.; Shen, T.; Khashabi, D.; and Choi, Y. 2023. Generating Sequences by Learning to Self-Correct. In *International Conference on Learning Representations* (ICLR).

West, P.; Lu, X.; Dziri, N.; Brahman, F.; Li, L.; Hwang, J. D.; Jiang, L.; Fisher, J.; Ravichander, A.; Chandu, K.; Newman, B.; Koh, P. W.; Ettinger, A.; and Choi, Y. 2023. The Generative AI Paradox: "What It Can Create, It May Not Understand". *CoRR*.

Wilcoxon, F. 1945. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6): 80–83.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *AAAI*, 2852–2858. AAAI Press.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-Rewarding Language Models. *CoRR*.

Zheng, L.; Chiang, W.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E. P.; Zhang, H.; Gonzalez, J. E.; and Stoica, I. 2023a. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *CoRR*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023b. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems* (NeurIPS).