Pride and Prejudice: LLM Amplifies Self-Bias in Self-Refinement

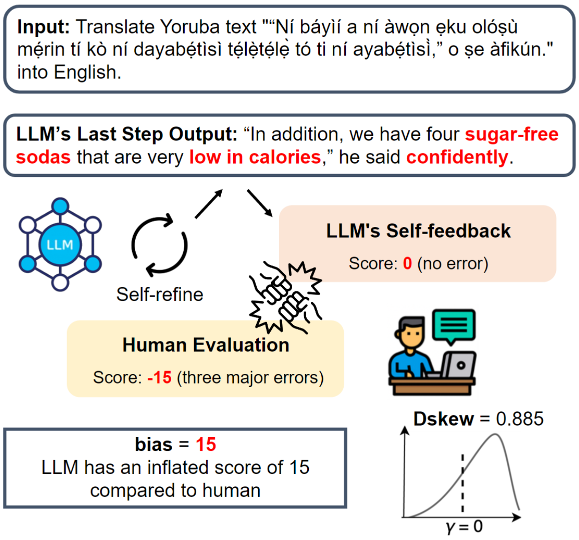
Wenda Xu[†], **Guanglei Zhu**[‡], **Xuandong Zhao**[†], **Liangming Pan**[†],
Lei Li[‡], **William Yang Wang**[†]
[†]University of California, Santa Barbara, [‡]Carnegie Mellon University
`{wendaxu,xuandongzhao,liangmingpan,william}@cs.ucsb.edu,`
`{guanglez,leili}@cs.cmu.edu`

Abstract

Recent studies show that large language models (LLMs) improve their performance through self-feedback on certain tasks while degrade on others. We discovered that such a contrary is due to LLM's bias in evaluating their own output. In this paper, we formally define LLM's self-bias – the tendency to favor its own generation – using two statistics. We analyze six LLMs (GPT-4, GPT-3.5, Gemini, LLaMA2, Mixtral and DeepSeek) on translation, constrained text generation, and mathematical reasoning tasks. We find that self-bias is prevalent in all examined LLMs across multiple languages and tasks. Our analysis reveals that while the self-refine pipeline improves the fluency and understandability of model outputs, it further amplifies self-bias. To mitigate such biases, we discover that larger model size and external feedback with accurate assessment can significantly reduce bias in the self-refine pipeline, leading to actual performance improvement in downstream tasks. The code and data are released at `https://github.com/xu1998hz/llm_self_bias`.

1 Introduction

Large language models (LLMs) have shown strong capabilities in many NLP tasks. While these models still make mistakes, recent studies show that "self-refine" (also known as "self-reflection") is promising to rectify errors based on LLM's self-feedback (Madaan et al., 2024; Chen et al., 2024; Shinn et al., 2024; Manakul et al., 2023; Pan et al., 2023). Meanwhile, opposite study also shows that LLMs fail to correct their mistakes and their performance even gets worse after self-feedback (Huang et al., 2023b). These contradictory results suggest that LLM's self-feedback is unreliable. Self-refine procedure relies on LLM's evaluation capability of the generated text. We hypothesize that if there is a bias during the self-evaluation process, such bias will be amplified during iterative self-refinement. This is consistent with a prior finding that LM-based metrics (e.g. BARTScore) exhibit "narcissism" during self-evaluation, *i.e.*, the metric model favors text generated by the same underlying language model in the context of summarization tasks (Liu et al., 2023b). However, it remains unclear whether bias exists universally in LLMs across a wide range of tasks. How to quantify such biases? How does this "narcissism" impact LLM's self-refinement?



Input: Translate Yoruba text ""Ní báyìí a ní àwọn ẹku olóṣù mẹ́rin tí kò ní dayabẹ́tìsì tẹ́lẹ̀tẹ́lẹ̀ tó ti ní ayabẹ́tìsì," o ṣe àfikún." into English.

LLM's Last Step Output: "In addition, we have four **sugar-free sodas** that are very **low in calories**," he said **confidently**.

Figure 1: How LLM's self-feedback inflates scores compared to human assessment. Bias is the mean difference between LLM and human scores, while skewness (Dskew) measures the asymmetry of their distribution around zero. Non-biased estimation will have Dskew=0.

In this work, we define "self-bias" to the degree that an LLM favors its own generation. We propose to use two principled statistics to estimate self-bias in LLM's self-refinement procedure. The first one measures the degree of inflation in the LLM's self-evaluation compared to the true (human) evaluation. The second measures whether LLM's self-evaluation is skewed compared to the ture estimate. Figure 1 illustrates these two statistics. We examine self-bias scores on six diverse LLMs, covering four languages across three distinct tasks: machine translation, constrained text generation, and mathematical reasoning. We find that self-bias is universal in self-refine and self-rewarding pipelines, regardless of the languages and tasks. This bias causes LLMs to optimize for false positive corrections rather than improving the actual output quality.

We further investigate what is the real benefit of self-refine. We find that while the self-refine pipeline improves the fluency and understandability of model outputs, it does not necessarily lead to intended improvements as specified in the prompt. Moreover, LLMs may favor texts that mirror their style, potentially leading to false positive optimization and reduced diversity in text generation. To mitigate the self-bias, we propose two solutions: increasing the model size and incorporating external feedback to provide accurate assessment, thereby directing the LLM towards more accurate self-correction. Our contributions are:

1. We formally define the self-bias of an LLM using two principled estimated statistics.
2. We quantify self-biases for six diverse LLMs and find that self-bias amplifies during self-refine across many languages and tasks.
3. We observe two factors that contribute to self-bias and pinpoint two directions to mitigate it and elicit LLMs' self-correction ability.

2 Related Work

Large Language Model Self-correction. Recent works demonstrate that LLM can utilize its own feedback signal to refine itself (Madaan et al., 2024; Chen et al., 2024; Shinn et al., 2024). Wang et al. (2023) further proposed to sample diverse reasoning paths and use a majority vote to find the most confident answer. Huang et al. (2023a) leverages self-consistency to further fine-tune the LLM on the most confident reasoning path with diverse instruction formats. On the other hand, LLM's self-feedback can also be used as a reward signal to further align LLM to follow instructions (Gulcehre et al., 2023; Yuan et al., 2024).

Despite some demonstrations of performance improvements, most findings indicate that LLMs struggle to rectify their initial mistakes, and their performance even worsens after self-correction (Huang et al., 2023b; Tyen et al., 2023; Ke et al., 2023). This issue arises because the quality of the model's self-generated feedback is bounded by its existing knowledge and abilities (Stechly et al., 2023; Hong et al., 2023). Therefore, internal feedback may not offer any extra advantage for improving the results; it might even steer the model away from the correct answer (Valmeekam et al., 2023). However, prior works only had empirical observations on this phenomenon, while lacking a quantitative analysis. Moreover, prior works only focus on specific tasks, such as reasoning or code generation. In this work, we are the first to quantitatively analyze the self-bias of different LLMs across three tasks and four languages, which provides a novel and generalizable view to address the perils of self-refine.

LLMs as Evaluators. Liu et al. (2023a) leverages GPT-4 to evaluate text through chain-of-thoughts prompting. Fu et al. (2023) leverages GPT-3's sequence likelihood to estimate model performance. Kocmi and Federmann (2023); Xu et al. (2023) designed detailed error schemes for LLM to output fine-grained error annotations. Despite the popularity of using LLMs as evaluators, Koo et al. (2023) pointed out that LLM exhibits cognitive bias when evaluating the text, misaligning from human preference. Zheng et al. (2023) pointed out LLMs have verbosity and self-enhancement bias, which makes them prefer long and verbose answers and answers generated by themselves. Chang et al. (2023) found out that LLM prefers memorized text over non-memorized text, creating unfair judgments over texts. Deutsch et al. (2022); Liu et al. (2023b) point out that reference-free metrics are inherently biased on their own outputs.

Although the above empirical studies provide valuable insights, they lack a formal definition to quantify those biases nor provide a connection to the self-refine framework. In this work, we define and quantify self-bias and provide the first in-depth analysis of its impact on the self-refine pipeline. We analyze potential bias attributions and pinpoint two mitigation directions.

3 Quantifying Self-Bias

This section outlines the approach used to quantify the self-bias exhibited by LLMs in an iterative self-refinement pipeline. We employ statistical bias and distance skewness (Szekely and Móri, 2006) estimation to measure self-bias.

3.1 Iterative Self-Refinement in LLMs

Self-refinement is an inference time method, in which the LLM first generates a response y_i to a given prompt x, and then the same LLM generates feedback f_i based on the candidate output y_i and input x. Based on feedback f_i, input x, and candidate output y_i, the LLM then generates a refined output r_i. LLM iterates between the feedback and the refinement steps, continuing until it reaches a predetermined number of iterations. At each refinement step, the refined output will only be accepted if it demonstrates superior quality compared to the previously generated text. The quality of the text is assessed through self-feedback from the language model itself. At each feedback or refinement step, LLM only sees the last iteration's generation or feedback, without accessing the entire history of output or feedback.

3.2 Bias Estimation

We estimate the self-bias of LLMs using the statistical bias definition. This bias is characterized by the disparity between an LLM's predicted quality score and the expected quality score, as follows:

$$\text{Bias}(\hat{\theta}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbb{E}[\hat{\theta}_i] - \theta_i), \quad (1)$$

where $\mathbb{E}[\hat{\theta}_i]$ is an expected LLM's quality prediction at sample i, and θ_i denotes the true quality of sample i. Ideally, θ_i should be derived from human annotations, for example, multidimensional quality metrics (MQM) human annotations (Freitag et al., 2021) for machine translation, or predefined criteria such as word coverage for constrained text generation (Madaan et al., 2024). The LLM's quality prediction is expected to precisely follow the human annotation procedure or predefined criteria, ensuring consistency between θ and $\mathbb{E}[\hat{\theta}]$. When $\text{Bias}(\hat{\theta}) > 0$, the LLM assigns a higher quality score to its own sample compared to the expected quality score. When $\text{Bias}(\hat{\theta}) < 0$, the LLM underestimates the sample quality compared to the expected quality score. The larger the value of $\text{Bias}(\hat{\theta})$, the more pronounced the LLM's bias against its own samples.

3.3 Distance Skewness Estimation

In an ideal scenario, an unbiased LLM should have equal chance of over-estimation and underestimation of text quality ($\text{Bias}(\hat{\theta}) = 0$), resulting in a perfectly symmetric distribution when plotting $\mathbb{E}[\hat{\theta}] - \theta$. However, $\text{Bias}(\hat{\theta}) = 0$ does not guarantee a symmetric distribution (In Figure 2, one tail could be long and thin, while the other is short and fat, yet they balance out overall). Therefore, we introduce another meta-metric, distance skewness, to measure the asymmetry of $\mathbb{E}[\hat{\theta}] - \theta$'s distribution. Specifically,

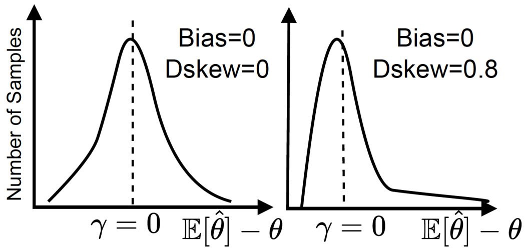


Figure 2: $\text{Bias}(\hat{\theta}) = 0$ does not guarantee a symmetric distribution of $\mathbb{E}[\hat{\theta}] - \theta$. One tail could be long and thin, while the other is short and fat (shown in the right figure). We use distance skewness to measure the asymmetry of distribution. Therefore, using two meta-metrics as complimentary, we can measure the self-bias of LLM.

$$d\text{Skew}_n(X) = 1 - \frac{\sum_{i,j}\|x_i - x_j\|}{\sum_{i,j}\|x_i + x_j - 2\gamma\|}, \quad (2)$$

where x_i and x_j are two independent identical random examples drawn from $\mathbb{E}[\hat{\theta}] - \theta$. $d\text{Skew}_n(X)$ measures the asymmetry of X with respect to γ. Distance skewness ranges between 0 and 1. $d\text{Skew}_n(X)$ equals 0 if and only if X is diagonally distributed respect to γ. $d\text{Skew}_n(X)$ equals 1 if and only if X is distributed at a constant on one side of γ. A higher distance skewness indicates a more asymmetric distribution of $\mathbb{E}[\hat{\theta}] - \theta$. In our experimental setup, we use both bias and distance skewness to measure the model's bias towards its quality prediction.

4 Analyzing LLM's Self-Bias

4.1 Experimental Setup

We include three closed-source LLMs (GPT-4 (Achiam et al., 2023), GPT-3.5-Turbo and Gemini (Team et al., 2023)) and three open-source LLMs (LLaMA2-7B (Touvron et al., 2023), Mixtral-MOE 8x7B (Jiang et al., 2024) and DeepSeekMoE 16B (Dai et al., 2024)). These models have been shown to have strong instruction-following capabilities (Madaan et al., 2024; Shinn et al., 2024), making them well-suited to demonstrate self-bias.

For each model, we first prompt it to produce the